©1993 LEGO Group.

Logo Core is ©LCSI 1993.

The sentence "Do you byte when I knock?" and "Just a bit off the block!" are ©LEGO Group 1992.

Student activities, teacher notes and reference pages may be photocopied for educational use within purchasing institutions. All other rights reserved.

LEGO DACTA® Control Lab Technology Investigations and Inventions ISBN 0-914831-90-9

LEGO DACTA® Control Lab Literature Pack ISBN 0-914831-98-4

®LEGO and LEGO DACTA are registered trademarks of INTERLEGO AG.

Control Lab™ and Building Technology for the Next Generation™ are exclusive trademarks of INTERLEGO AG.

IBM is a registered trademark of International Business Machines Corporation. MS-DOS is a registered trademark of Microsoft Corporation. Apple, the Apple logo and Macintosh are registered trademarks of Apple Computer, Inc. Tandy is a registered trademark of Tandy Corporation. LCSI is a registered trademark of Logo Computer Systems, Inc.

Distributed in the USA by:

LEGO Dacta
LEGO Systems, Inc.
555 Taylor Road
Enfield, CT 06083-1600
(800) 527-8339

Distributed in Canada by:

LEGO Dacta
LEGO Canada, Inc.
380 Markland Street
Markham, Ontario L6C 1T6
(800) 387-4387

Acknowledgements

LEGO Dacta gratefully acknowledges the contributions of the following:

Writing	Thomas Barrowman, Queensbury, NY Richard Crandall, Lexington, MA Michael Muir, Skowhegan, ME
School Testing	Brian Boucher, John F. Kennedy Middle School, Enfield, CT Paul Murphy, Dr. Arthur Sullivan Middle School, Worcester, MA Frank Riddle, An Wang School, Lowell, MA Jolane Roy, An Wang School, Lowell, MA James Scarth, An Wang School, Lowell, MA Arthur Searle, Exeter-W. Greenwich Junior / Senior High School, W. Greenwich, RI Michael True, Dr. Arthur Sullivan Middle School, Worcester, MA
Software Testing	Christine Appert, Charlottesville, VA Glen L. Bull, Charlottesville, VA Gary Stager, Wayne, NJ Vicky Zimmer, Buffalo, NY
Reviewing	Steve Barbato, Dover, DE John Brown, LEGO Dacta Patricia Hutchinson, Philadelphia, PA William Kabakjian, Philadelphia, PA Ron D. Todd, Philadelphia, PA
Design and Layout	The Super Market, East Longmeadow, MA
Illustration	Rodney Goudreau, Technical Illustrator Paul Metcalf, Illustrator
Photography	John Stoppel, Stoppel Photography
Consulting Editor	Tom Lough, LEGO Dacta
Project Direction	Catherine Helgoe, LEGO Dacta

Printed in the U.S.A.

822206

Technology Investigations and Inventions

Before students use this book, make sure they have:

___ Installed the LEGO DACTA® Control Lab™ hardware and software. (See pages 6-10 in the *Setup Guide* or pages 4-5 of the *Quick Start Guide.*)

___ Organized Technology set #9701. (See pages 2-3 of blue building card #9701.)

___ Completed the introductory explorations. (If they are new users, see pages 12-40 of the *Setup Guide and Introductory Explorations* book. If they are familiar with the computer and LEGO DACTA products in general, go to the activities in the *Quick Start Guide.*)

Before using this book with your class, make sure you have:

___ Selected the activities and supplemental materials on which you want students to work.

___ Made photocopies of the student activity pages and supplemental materials for each student or student group.

Routes Through the LEGO DACTA Control Lab Materials

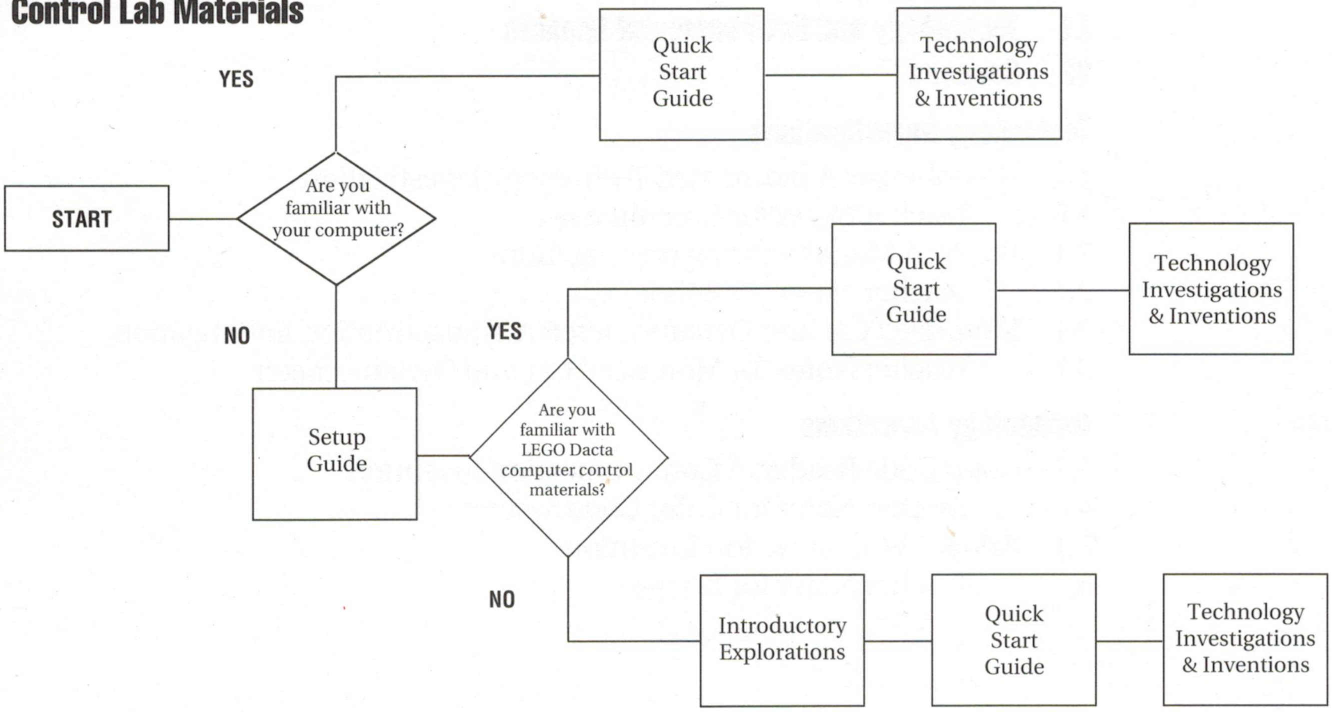

YES
Quick Start Guide
Technology Investigations & Inventions
START
Are you familiar with your computer?
Quick Start Guide
Technology Investigations & Inventions
NO
YES
Setup Guide
Are you familiar with LEGO Dacta computer control materials?
NO
Introductory Explorations
Quick Start Guide
Technology Investigations & Inventions

Table of Contents

4 Overview of the Curriculum Materials
5 Description of Student Investigation and Invention Pages
6 Description of Teacher Notes
7 Classroom Management Questions and Answers
9 What Do Students Learn?
9 Technology
10 Science
11 Mathematics
12 Language Arts
12 Social Studies
13 Process and Inquiry Skills
13 Critical Thinking Skills
14 Assessment Ideas
15 Approaches to Problem Solving
15 A "Conceptual Framework" Approach
16 A Systems Approach
18 What Are Systems?
18 Open Loop
19 Program Control
19 Closed Loop
21 Technology and Environmental Impacts
22 Careers

Technology Investigations

1.1 Greenhouse: A Bio-related Technology Investigation
1.5 Teacher Notes for Greenhouse
2.1 Robot: A Manufacturing Investigation
2.4 Teacher Notes for Robot
3.1 Motorized Car and Dynamometer: A Transportation Investigation
3.7 Teacher Notes for Motorized Car and Dynamometer

Technology Inventions

4.1 Color Code Reader: A Communication Invention
4.6 Teacher Notes for Color Code System
5.1 Bridge: A Construction Invention
5.5 Teacher Notes for Bridge

6.1 Supplemental Materials

6.2 Technology Set Parts List
6.6 Quick Reference Sheets
6.10 Measurement Worksheet
6.11 Front View Graph Paper
6.13 Top View Graph Paper
6.15 Problem Solving steps for the Conceptual Framework Approach
6.17 Project Evaluation Checklist
6.19 Systems Problem Solving Model
6.20 Systems flowcharts
  6.20 Open
  6.20 Program
  6.21 Closed
6.22 Bibliography

Overview of the Curriculum Materials

Technology education brings together knowledge gained from other disciplines as the basis for studying systems in the areas of:

Bio-related technology
Communication
Production / Construction
Production / Manufacturing
Transportation

Problem solving activities in these five areas help to form the foundation of technology education. Students use the LEGO DACTA® Control Lab™ system as their materials for designing , building and programming solutions to problem solving situations described in a design brief.

Control Lab curriculum materials are organized to build student experience in three steps.

1) First, students EXPLORE. Students encountering new materials, equipment or concepts need to become familiar with them. Through the exploration, students are introduced to the uses and capabilities of the LEGO DACTA Control Lab equipment.

   The *Setup Guide and Introductory Explorations* book and the *Quick Start Guide* include exploration activities.

2) Once they are familiar with the materials, students INVESTIGATE. While engaged in an investigation activity, students learn specific concepts and develop skills. For example, students follow instructions to build a greenhouse model, then program a control system to monitor the temperature inside.

   Three investigation activities are included in this book, *Technology Investigations and Inventions.* They are as follows:

Investigation	**Technology Area**	**Building Instruction #**
Greenhouse	Bio-related technology	9701-1
Robot	Production/ Manufacturing	9701-7
Motorized Car and Dynamometer	Transportation	9701-3

✏ **Teacher note:**

For all activities, see the "What do students learn?" section in this book for how each activity can relate to specific concepts in technology, science, math, language arts and social studies.

3) After students learn new concepts and develop new skills with materials, they are ready to apply what they know. They can INVENT their own solutions to a problem situation. For example, students can design a bridge to allow traffic to cross a river. Then they must build a working solution of their design and program it to function. No step-by-step instructions are provided for the student; however, a solution is shown in the teacher notes.

   Two invention activities are included in this book, *Technology Investigations and Inventions.* They are as follows:

Investigation	Technology Area	Building Instruction
Color Code Reader	Communication	Sample solution included in teacher notes
Bridge	Production/ Construction	Sample solution included in teacher notes

Description of Student Investigation and Invention Pages

The student pages for each of the five technology investigations and inventions include the following sections:

Background Information	Provides information about the technology areas, presents concepts and defines key vocabulary.
Define the Problem and Problem Statement	Illustrates and describes the problem.
Design Brief	Lists specific building and programming criteria for solving the problem. Additional criteria suggest other features to add to the design solution for more building and programming challenges.
Building Ideas	Suggests one way to build a solution to the design brief.

Programming Ideas	Suggests one or more ideas to think about when developing a programming solution for the design brief.

Description of Teacher Notes

✏ **Teacher note:** **More specific concepts in technology, math, science, language arts and social studies as they relate to each activity are shown in the "What do students learn?" section in this book.**

Objectives	Describes general knowledge the students acquire by completing the activity.
Materials Needed	Lists essential and optional materials students can use to complete the activity.
Background Information	Includes additional remarks related to this student section.
Define the Problem	Includes additional remarks related to this student section.
Building Ideas	For Investigation activities, students have step-by-step building instructions in the Technology set (#9701). For the Invention activities, students are given no building instructions; however, the teacher notes show one solution for each activity.
Programming Ideas	Shows a solution for the main criteria and, if appropriate, solutions for the additional criteria.
Related Concepts and Extensions	Suggests activities or discussion points to cover science, math, language arts, social studies and additional technology concepts (interdisciplinary).

Classroom Management: Questions and Answers

✏ **Teacher note:**

To manage the student software project files, format and label separate disks for each student group to store its projects. Or, create a subdirectory (MS-DOS) or folder (Macintosh) on a hard disk for each student group.

Q. How can I effectively manage the building elements in the LEGO DACTA® Technology set?

A. The most important thing to do is to involve the students in managing the sets responsibly. Once they realize that any missing elements can limit their building activities, most students will put forth a reasonable effort to help keep track of the sets.

- Mark each Technology set with an identification number.
- Require students to sign the sets in and out.
- Establish a "Lost and Found" cup for stray building elements.
- Ask students to check the floor for elements at the end of class.
- Make sure students have adequate time to check and store their sets.
- Assign a daily "captain" to check sets upon their return.
- Consider purchasing an additional Technology set for spare parts.

Q. What should I do if some Technology set building elements are lost or broken?

A. If you require any kind of service for your Technology set, please write to Susan Williams, Consumer Affairs, LEGO Systems, Inc. P. O. Box 1138, Enfield, CT 06083-1138 or call (203) 763-3211.

Q. How can I deal with students who have not finished building their models and do not want to start over the next time the class meets?

A. The activities in the *Setup Guide and Introductory Explorations* book and some of the activities in the *Quick Start Guide* can be accomplished within one class period. Most of the investigations and inventions in this book, however, require several class periods, especially if students are designing their own models as a solution to the design brief. If you are using the #9701 Technology building sets for more than one class at one time, students will not be able to save their partially assembled models. This could lead to some frustration. Where possible, schedule the building sets for one class at a time or arrange longer periods for your students.

Q. How can I arrange for inservice training?

A. Training is always an important consideration, especially when dealing with a product with as many components as LEGO DACTA Control Lab. LEGO Dacta offers inservice workshops in many parts of the United States and Canada. For more information, call LEGO Dacta USA at (800) 527-8339 or LEGO Dacta Canada at (800) 387-4387.

Q. Can I use other materials with LEGO DACTA® Control Lab?

A. Of course. This could make the activities more interesting and realistic. Most likely you already have paper, string, cardboard and similar materials in your classroom. Here is a partial listing of other items which might be handy:

- Popsicle sticks
- Thin cotton cloth or other material
- Small plastic bags
- Paper cups
- Corks
- Illustration board

Q. What other equipment do you recommend?

A. You may wish to have available for students on a request basis other items, such as the following:

- Stroboscope
- Ruler or meter stick
- Set of small weights or masses
- Force meter or spring scale
- Protractor

✏ **Teacher note:**

If you do not have the current brochure, contact LEGO Dacta USA at 1 (800)-527-8339 or LEGO Dacta Canada at (800) 387-4387 and request one.

Additionally, you may wish to purchase LEGO® elements to extend the LEGO DACTA Control Lab design possibilities. See the current LEGO Dacta brochure for information about:

- Extra elements including, sensors, motors, lights, sound elements, bricks, beams, gears, tires and axles.
- Other specialty elements such as pneumatics, pumps and battery boxes.

Q. How do I help students learn more about Logo programming?

A. Here are some ideas:

- Refer students to the Help text in the Control Lab software menu for definitions and examples of the Control Lab primitives (commands and reporters) and error messages.
- Read explanations and examples in the *Reference Guide.* Review the programming examples in the Programming Ideas section of the teacher notes and the Quick Reference sheets in the Supplemental Materials section of this book.
- See the Bibliography in the Supplemental Materials section of this book for a listing of other Logo resources.

What Do Students Learn?

The grids shown on the next few pages summarize the technology, science, math, language arts and social studies concepts that students can learn in each activity.

Technology

	Activities				
Concepts	**Bridge**	**Dynamometer**	**Greenhouse**	**Robot Arm**	**Code Reader**
Biotechnology			X		
Careers	✓	✓	✓	✓	✓
Choosing Resources	●✓	●✓	●✓	●✓	●✓
Communication					X
Environmental Impacts					
Natural	R	●	●	●	●
Human made	R	●	●	●	●
Human Values and Attitudes	R	●	●	●	●
Problem Solving	X	X	X	X	X
Production/Construction	X				
Production/Manufacturing				X	
Systems					
Closed Loop	●	●	●	●	●
Feedback (Input From)					
Angle	X	X	X	X	R
Light	R	X	R	R	X
Temperature			X		
Touch			R		
Open Loop	●	●	●	●	●
Output					
Sound Element	R				X
Lamp	R	X	X		R
Motor	X	X	X	X	R
Program Control	●	●	●	●	●
Transportation	R	X			

X = major concept

✓ = relates to the Conceptual Framework Approach to Problem Solving

● = relates to the Systems Approach to Problem Solving

R = idea in the teacher note section called Related Concepts and Extensions

Science

	Activities				
Concepts	**Bridge**	**Dynamometer**	**Greenhouse**	**Robot Arm**	**Code Reader**
Color					X
Electricity	X	R	X	X	X
Electromagnetism		R			
Energy transformation/ conservation	X	X	X	X	X
Forces	X	X	X	X	X
Forms of Energy					
Heat	R	X	X		
Kinetic	X	R	X	X	
Light	R	X	X		X
Sound	R	✓		X	X
Friction	X	X	X	X	
Mass/weight	X	R	X	X	
Mechanical advantage	X	X	X	X	
Plant Growth			R		
Process Skills					
Collecting data	R	X	X	X	X
Designing experiments	R	R	X		X
Hypothesizing	R	R	X		X
Making conclusions	R	R	X	X	X
Observing	X	X	X	X	X
Predicting	X	X	X	X	X
Reactions/interactions	X	X	X	X	X
Rotational Dynamics	X	X	X	X	
Simple Machines	X	X	X	X	
Velocity		X		X	
Work and Power	X	X		X	

X = major concept

✓ = relates to the Conceptual Framework Approach to Problem Solving

● = relates to the Systems Approach to Problem Solving

R = idea in the teacher note section called Related Concepts and Extensions

Mathematics

Activities

Concepts	Bridge	Dynamometer	Greenhouse	Robot Arm	Code Reader
Absolute Value	X	X			
Addition	X	X	X	X	X
Algebra	X	X	R	X	X
Converting Units	X	X	X	X	X
Decimals	X	X	X	X	X
Division	X	X	X	X	X
Estimation	X	X	X	X	X
Fractions	X	X	X	X	X
Graphing			X		
Greater than / Less than	X		X	X	X
Measuring					
Angle		X	X	X	R
Circumference		X			
Distance	X	X		X	X
Mass	X	X			
Rotation	X	X		X	
Temperature			X		
Time	X	X	X	X	
Velocity		X			
Multiplication	X	X	X	X	X
Percent		X			X
Ratios	X	X	X	X	X
Rounding	X	X	X		
Subtraction	X	X	X	X	X
Trigonometry		X			
Variables			X	X	R

X = major concept

✓ = relates to the Conceptual Framework Approach to Problem Solving

● = relates to the Systems Approach to Problem Solving

R = idea in the teacher note section called Related Concepts and Extensions

Language Arts

Activities

Concepts	**Bridge**	**Dynamometer**	**Greenhouse**	**Robot Arm**	**Code Reader**
Creative writing				R	
Following directions	X	X	X	X	X
Grammar	X	X	X	X	X
Oral presentations		R			R
Reading directions	X	X	X	X	X
Recording data	X	X	X	X	X
Spelling	X	X	X	X	X
Technical writing	R	R	R	R	R
Typing	X	X	X	X	X
Using Media					
Graphical	X	X	X	X	X
Textual	X	X	X	X	X
Verbal	X	X	X	X	X
Vocabulary	X	X	X	X	X
Writing Reports	R	R	X	X	R

Social Studies

Activities

Concepts	**Bridge**	**Dynamometer**	**Greenhouse**	**Robot Arm**	**Code Reader**
Demographics			R		R
Economics	R	R	R	R	
Government	R				R
History				R	
Political Science	R				R
Psychology			R		
Sociology			R		

X = major concept

✓ = relates to the Conceptual Framework Approach to Problem Solving

● = relates to the Systems Approach to Problem Solving

R = idea in the teacher note section called Related Concepts and Extensions

Mathematics

Activities

Concepts	**Bridge**	**Dynamometer**	**Greenhouse**	**Robot Arm**	**Code Reader**
Absolute Value	X	X			
Addition	X	X	X	X	X
Algebra	X	X	R	X	X
Converting Units	X	X	X	X	X
Decimals	X	X	X	X	X
Division	X	X	X	X	X
Estimation	X	X	X	X	X
Fractions	X	X	X	X	X
Graphing			X		
Greater than / Less than	X		X	X	X
Measuring					
Angle		X	X	X	R
Circumference		X			
Distance	X	X		X	X
Mass	X	X			
Rotation	X	X		X	
Temperature			X		
Time	X	X	X	X	
Velocity		X			
Multiplication	X	X	X	X	X
Percent		X			X
Ratios	X	X	X	X	X
Rounding	X	X	X		
Subtraction	X	X	X	X	X
Trigonometry		X			
Variables			X	X	R

X = major concept

✓ = relates to the Conceptual Framework Approach to Problem Solving

● = relates to the Systems Approach to Problem Solving

R = idea in the teacher note section called Related Concepts and Extensions

Process and Inquiry Skills

In addition to the specific concepts, process and inquiry skill development is at the heart of any hands-on instruction. Developing these skills are also outcomes of the Control Lab projects. Outcomes are the skills students retain after the instruction is delivered.

Observing the behavior of objects and their interactions.
Describing clearly what is observed.
Identifying important criteria and variables in an investigation.
Measuring distance, circumference, speed and time accurately.
Recognizing space and time relationships.
Classifying objects or concepts into categories.
Building and using models to investigate and experiment.
Collecting and interpreting data from investigations.
Comparing and contrasting information from different sources.
Constructing personal theories from data.
Predicting the behavior of an object.
Hypothesizing a possible relationship between two variables.
Experimenting to confirm or disprove a hypothesis.
Questioning or pondering curiously a supposition.
Inferring or deducing facts from available information.
Formulating and solving problems.
Replicating an experiment or confirming a law.

Critical Thinking Skills

The activities in this book also promote critical thinking skills. The following are samples of different types of questions designed to help students develop these skills.

Thinking Skills	Sample Questions
Recalling	What is the name of this LEGO® element?
Understanding	How does the LEGO fan model demonstrate speed?
Applying	How could you make a faster motorized vehicle?
Analyzing	How are the best bridge designs alike?
Synthesizing	In what other ways will these elements fit together?
Evaluating	Which device has the most effective design?

✏ Teacher note:

Process skills and critical thinking skills relate closely to the types of skills employers need from employees. See the section on Careers for more information.

Assessment

In general, assessment can serve several purposes. It can:
- Provide a written record of progress.
- Give feedback to students.
- Provide students with an opportunity for self-assessment.

Using a wide variety of assessments will provide a broader picture of student progress. For example, intellectual skills and knowledge (cognitive area) might be assessed by pencil and paper tests or student projects and reports. Changes in student interests, values, and attitudes (affective area) might be indicated by the results of frequent surveys or journal entries. Manipulative skills (psychomotor area) could be indicated by the results of building challenges or by frequent teacher observation.

Assessments focus not only on what individual students can remember, but on what they can understand and apply as well. See Process and Inquiry Skills and the Critical Thinking Skills sections on previous pages for examples of different types of questions to consider.

Here are some assessment ideas to consider for activities in this book.

✏ **Teacher note:**

For students' drawings, you can copy the LEGO graph paper in the Supplemental Materials section of this book.

- A journal or log could be part of the assessment for each team of students or each individual student. The LEGO DACTA® Control Lab software can be used for documenting the projects.
- Question individual students randomly during building activities to determine if they understand operational definitions and concepts. Encourage students to ask each other questions also.
- Consider performance-based assessments in which students individually or in teams build a model which operates in a specified manner.
- Use the Project Evaluation Checklist in the Supplemental Materials section to track student progress.
- Use the Technology Set Parts List from the Supplemental Materials section to help improve students' vocabulary.

Approaches to Problem Solving

Few problems can be approached and solved according to some neat step-by-step arrangement. Designers may move from the stated problem to a sketch of an imaginative solution. Or, such analysis may spark an idea leading to immediate experimentation. Many problem solving steps may be used at the same time. For these reasons, several different formalized problem solving models are used today.

The following pages describe two approaches to problem solving:

1) A "Conceptual Framework" approach described in the International Technology Educators Association publication, *A Conceptual Framework for Technology Education* by Ernest Savage and Leonard Sterry.
2) A systems approach.

A "Conceptual Framework" Approach to Problem Solving

✏ Teacher note:

A student Problem Solving sheet following this approach is included in the Supplemental Materials section of this book. Use this form to help students work through the problem solving steps and connect the impacts and career implications. The Problem Solving sheet was developed by LEGO Dacta for use with Control Lab activities.

In the book, *A Conceptual Framework for Technology Education,* a broad approach to problem solving in technology education is described. The following steps are included:

1) Defining the problem.
2) Developing alternative solutions.
3) Selecting a solution.
4) Implementing and evaluating a solution.
5) Redesigning the solution.
6) Interpreting the solution (communicating to others how the solution works).

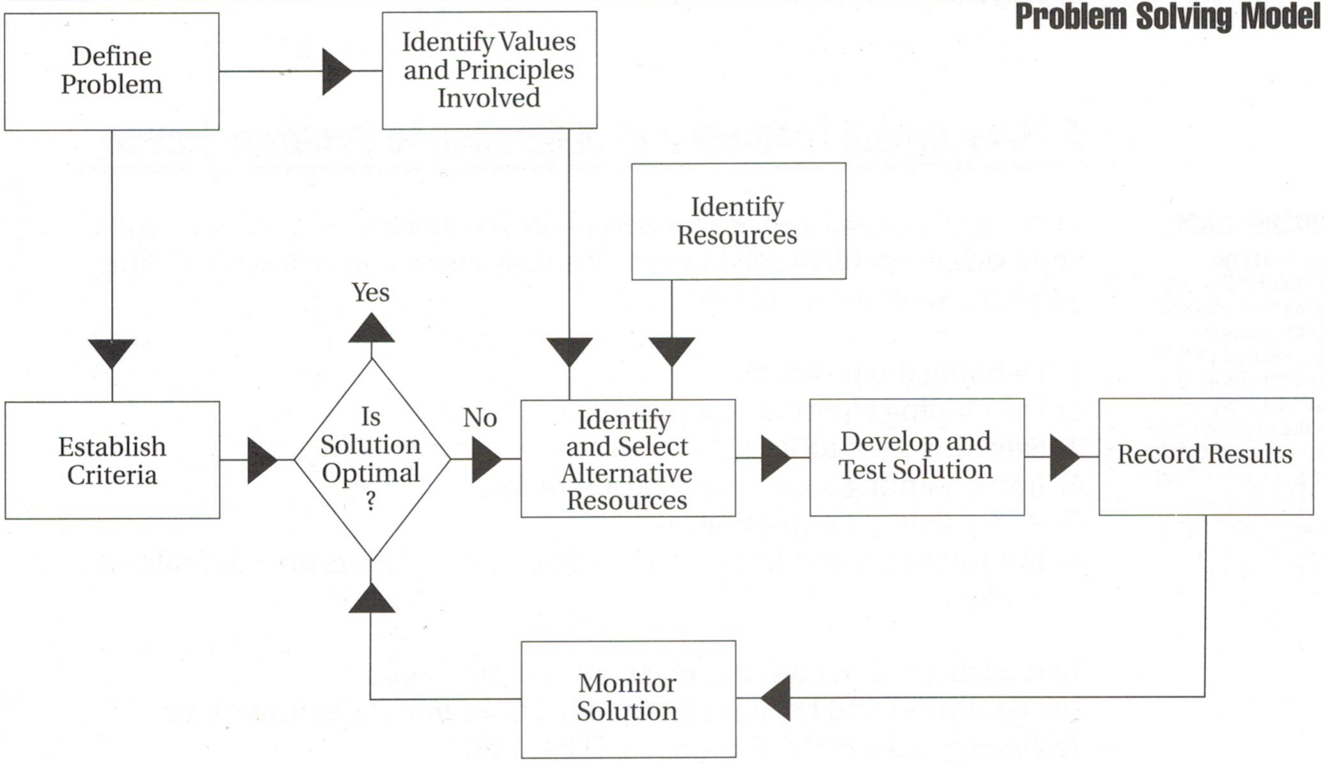
To read more about this approach, refer to the book:
Savage, Ernest and Leonard Sterry, ed. *A Conceptual Framework for Technology Education.* Reston, VA: ITEA, 1990.

A Systems Approach to Problem Solving

This problem solving approach is more structured and emphasizes the need to identify criteria and scientific principles before a solution is selected and evaluate impacts after the solution is tested. This problem solving model includes other features which make it especially appropriate for use in technology education. In particular, the organization of the model is keyed closely to the organization of the systems approach (described in the section following the problem solving model).

The systems approach is NOT intended to depict how the human brain functions; humans often think nonlinearly at many different levels and from several different perspectives within a short time period. The sequence of problem solving steps in the model is designed to organize the approach, not to restrict it.

Problem Solving Model



Study the chart starting with the block entitled, “Define Problem.” Each block is defined by a title.

Define Problem

Formulate a clear, simple, direct statement of the problem. Example: Build a fast self-propelled vehicle model.

✏ **Teacher note:**

A copy master of this chart is included in the Supplemental Materials section of this book.

Establish Criteria

The criteria (judgements, rules or standards) against which the solution is measured define when the problem solving process is finished. Example: The self-propelled vehicle must travel at least three feet per second and must not tip over.

Identify Values and Principles Involved

Students should clearly identify values relating to concerns such as environmental issues. They should also describe the scientific principles which relate to the problem. Example: The self-propelled vehicle should not pollute the environment. Principles of force, motion, friction, and inertia must be considered.

Identify Resources

Students should describe in detail each of the following seven resources: people, information, materials, tools and machines, capital, energy, and time. Here is an example of a problem involving a self-propelled vehicle.

People - problem solving team
Information - teacher, library
Materials - LEGO® building set
Tools and machines - computer
Capital - none required
Energy - electrical current
Time - classroom period

Identify and Select Alternative Solutions

Many techniques can be used to identify alternative solutions, including trial and error, inspiration and brainstorming. Students could make a list of many different solutions. One key is to identify characteristics which can be varied. Example: For the self-propelled vehicle, different solutions could have a short or long wheel base, light or heavy vehicle weight, three or more wheels, etc.

Develop and Test Solution

After discussion and perhaps some preliminary testing, students develop fully and test one solution. Example: Students build and test a long, lightweight, 4-wheel, self-propelled vehicle.

Record Results

Students record the results of the test. Example: The vehicle developed a speed of 3.2 feet per second and remained upright.

Monitor Solution

Students discuss the results and examine any impact on values described earlier. Example: The vehicle did not pollute the environment, but the wheels damaged the layer of topsoil.

Is Solution Optimal?

Students take both results and criteria into consideration to decide if the solution is the best one possible. If so, then the problem is considered solved. If not, they refine the solution or select another solution for development.

What Are Systems?

✏ **Teacher note:**

A systems approach to problem solving which relates closely to the closed loop system is described in the section A Systems Approach to Problem Solving in the book.

✏ **Teacher note:**

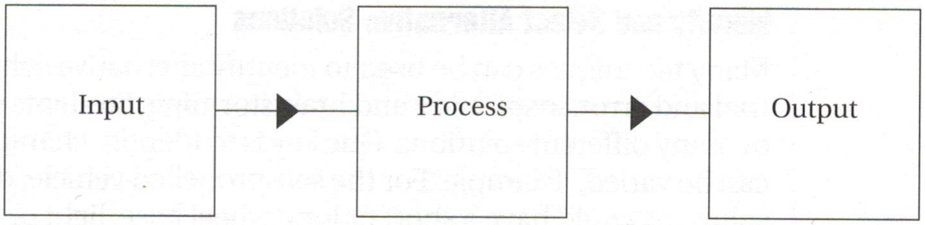
A description of LEGO DACTA® Control Lab as a system is explained on page 5 of the *Setup Guide*.

Each advancement in technology makes possible more complex technological systems which can generate new products to satisfy our wants and needs. These complex technological systems are developed from simpler systems.

Any technological system must have at least three components: input, process and output. **Input** consists of the commands and resources entered into a system. The commands describe the goal of the system, while the resources provide the wherewithal to accomplish that goal. **Process** refers to the action component; resources are combined to fulfill the desired system goal. The actual result of the process is called the output of the system. **Output** can be classified as desired or undesired and as expected or unexpected.

Open Loop

An open loop system is the simplest technological system. It consists of a sequence of input, process and output components. An open loop system cannot adjust or monitor itself. Control must be applied externally. Humans usually monitor open loop systems.

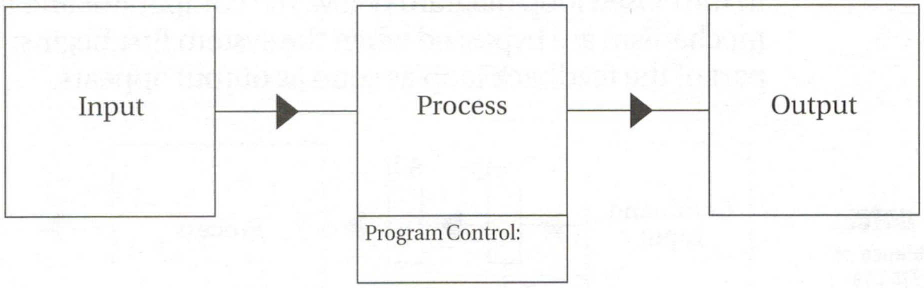


Example: A fan turns on when a touch sensor is pressed.

Command input:	Turn on fan motor when touch sensor is pressed.
Process:	Touch sensor provides an electrical path for powering the fan motor.
Output:	Fan turns on when sensor switch is pressed.

Program Control

A program control system is like an open loop system with a timer added. The system operation is performed for a set period of time or is repeated in a regular (or periodic) way. Like the open loop system, a program control system cannot adjust or monitor itself. All it can do is the task for which it was designed.



Example: A machine records and displays the RPM of a motorized car.

Command input:	Display the RPM of a car's back axle.
Process:	An angle sensor connected to the computer collects the number of axle rotations; time is also recorded by the computer. RPM is calculated and printed on the computer screen.
Output:	The RPM of the car shows on the screen.

Closed Loop

A closed loop system can:
- Evaluate its output and adjust its process accordingly.
- Operate without human intervention.
- Operate without regard to time.

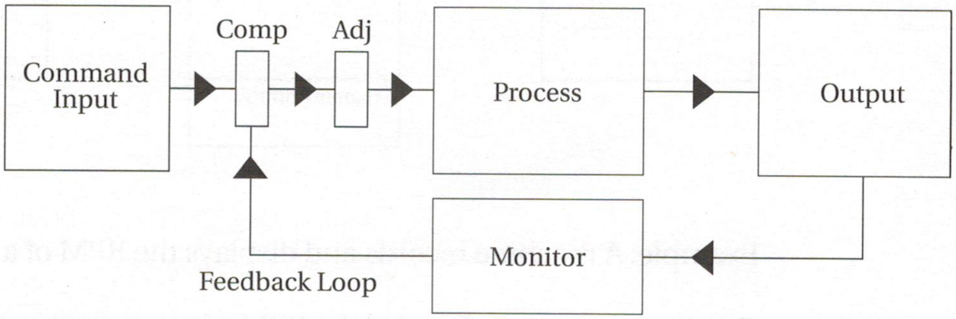
The diagram of a closed loop system resembles that of the open loop system, with the addition of a feedback loop. The feedback loop includes a monitor, a comparator and an adjustment mechanism.

The monitor is a specialized sensor which provides information about the nature of the output. The LEGO DACTA® Control Lab sensors provide feedback on temperature, light intensity, touch and rotation.

The Control Lab software includes a "monitor box" tool which displays the state of the sensor on the screen.

The comparator compares the monitor information about the output with information specified by the command input and makes a decision about whether the two values match sufficiently. The adjustment mechanism changes the process within a system so that the output results are closer to those specified. The computer is used as the comparator and the adjustment mechanism in the Control Lab system.

In the closed loop diagram below, the comparator and adjustment mechanism are bypassed when the system first begins; they operate as part of the feedback loop as soon as output appears.



✏ **Teacher note:**

Note the resemblance of this diagram to the systems approach to problem solving on page 16.

Example: A greenhouse vent is activated by a change in temperature.

Command input:	When there is no increase in temperature, maintain the position of the vent. When the temperature rises, open the vent.
Process:	Turn on the motor to open the vent. Electricity flows to vent motor.
Output:	The vent is open.
Monitor:	The sensor reports a change in temperature.
Comparator:	Interprets sensor report and decides if temperature is high enough to open the vent.
Adjustment:	If the temperature is high enough, the vent opens.

More complex systems made up of subsystems of the three fundamental diagrams are possible.

Technology and Environmental Impacts

In earlier times, technology consisted of finding ways to help humans obtain food, survive the extremes of cold and hot temperatures, and get from one place to the other alive. As technology developed, humans discovered and developed more effective methods of growing food, providing shelter, and traveling. Today, however, the process of technological advances is changing to take into account additional considerations relating to our environment.

✏ **Teacher note:**

Project WILD is a contributor to environmental awareness. Sponsored by state wildlife agencies and state departments of education, the project emphasizes the ecological effect technology has on wildlife and seeks to help us understand our responsibilities to all living things. See the Bibliography in the Supplemental Materials section of this book.

Outputs of technological systems are classified as desirable or undesirable and as expected or unexpected. Desirable and expected outputs are always the aim of a technological system. However, unexpected outputs sometimes occur. If the technological system was planned to take into account environmental concerns, the outputs should not produce a serious environmental impact.

Environmental concerns are focused on the undesirable outputs. The processing of undesirable and expected outputs can be planned in advance. For example, the waste products from nuclear power plants are definitely undesirable and expected, and must be processed with extreme caution. However, the most serious category is that of undesirable and unexpected outputs. Since the output was unexpected, no prior planning took place, presenting an opportunity for severe environmental damage. One example in this category has been the dumping of mercury into streams and rivers. No one anticipated the problem of mercury poisoning from fish which resulted.

As each LEGO DACTA® Control Lab activity is accomplished, students should be encouraged to discuss related environmental issues and categorize the possible outputs from their project. For example in the Greenhouse activity, students could discuss the impacts of adding insulation to the house on the temperature inside and the potential quality hazards of a house with too few air exchanges.

Careers

✏ **Teachers note:**

For more Information about the study, see the book *Workplace Basics: The Essential Skills Employers Want* by Anthony P. Carnevale, et.al., Jossey-Bass, Inc., San Francisco; CA, 1990.

Thanks to Judy Allen, LEGO Systems, Inc. for summarizing the general employment skills.

In a recent study conducted by the American Society for Training and Development and the U.S. Department of Labor, fifteen basic skills were identified as essential to workplace success. These skills relate closely to the process, inquiry and critical thinking skills identified in the Assessments section of this book. Working on LEGO DACTA® Control Lab projects can help students develop these skills which are important life habits.

General Employment Skills

Learning to learn	Learning effectively in any situation.
Reading	Acquiring information from printed or written symbols.
Writing	Combining physical production and mental processing to express ideas.
Computation	Performing basic mathematical functions.
Oral communication	Exchanging thoughts, ideas and messages.
Listening	Receiving, attending to, and assigning meaning to aural and visual stimuli.
Problem solving	Bridging a perceived gap between what is and what ought to be.
Creative thinking	Using different modes of thought to form new and relevant combinations of ideas.
Self esteem	Demonstrating a strong sense of self worth leading to a positive self concept.
Motivation and goal setting	Striving for successful performance in life and work situations.
Interpersonal skills	Getting along well with other people in most situations.
Teamwork	Relying on individual expertise and group collaboration to accomplish mutually agreed upon tasks and goals.
Negotiation	Facilitating a bargaining process between parties who have a perceived or actual conflict of interests.
Organizational effectiveness	Aligning one's behavior and attitudes with the philosophy and style of the organization.
Leadership	Empowering others.

Relating Careers to the Activities

Many of the general employment skills described on page 22 relate directly to the outcomes and skills explained in the section "What Do Students Learn?" in this book. The Bridge, Color Code Reader, Robot, Motorized Car and Dynamometer and Greenhouse are all LEGO DACTA® Control Lab projects that students can try. Each activity is from one of five technology areas and offers a chance for students to think about related careers.

Technology area	Activity	Related Careers
Bio-related Technology	The Greenhouse uses a temperature sensor to regulate the air temperature of a building. Passive solar energy is captured and computer controls maintain the temperature within a specified range.	Horticulturist, computer programmer, architect
Communication	The Color Code Reader uses a light sensing device. Light sensing technology is also used at supermarket checkouts and department stores.	Warehouse inventory clerk, computer systems training instructor
Construction	The Bridge uses bridge building concepts such as geometric shapes and angles, stress forces and structural design. Moveable bridges, such as the vertical lift design shown in the teacher notes for the activity, allow ships to pass through.	Structural engineer, civil engineer, bridgetender (operator)

Manufacturing	The Robot uses an angle sensor to identify position, mechanical gearing ratios, geometric angle relationships and mechanical design. Fifty years ago, robots were dreamed about. Now they are used to package milk and build cars.	Mechanical engineer, electronic engineer
Transportation	The Motorized Car and Dynamometer system is used to measure a car's speed or RPM (revolutions per minute). This is accomplished by using technology such as an angle sensor to measure rpm, visual display of the information, gear ratio design, wheel circumference and by varying its weight and shape.	Industrial designer, automotive repair service estimator, environmental analyst

As technology advances, new careers are created and others are eliminated. For this reason, education about technology and its impacts on the work force is an important and continuous endeavor.

To find more about careers, students can speak with the Guidance Department personnel or visit the library and ask for the book "Occupational Outlook Handbook" published by the U.S. Department of Labor, Bureau of Labor Statistics, Washington, DC 20212.

✏ **Teacher note:**

If students are following the "Conceptual Framework" approach to problem solving, the last section of the student Problem Solving copymaster in the Supplemental Materials section includes questions related to careers.

Greenhouse: A Bio-related Technology Investigation

Student Name: ______________________________ **Date:** ______________

Background Information

What is bio-related technology?

Bio-related technology activities offer ways of integrating scientific knowledge with technology to solve important problems. Food and fuel production, waste management and medical technology are areas that involve bio-related technology ideas.

In the greenhouse activity, you will learn about bio-related technology systems by designing, building and programming a solar heated greenhouse.

What is a greenhouse?

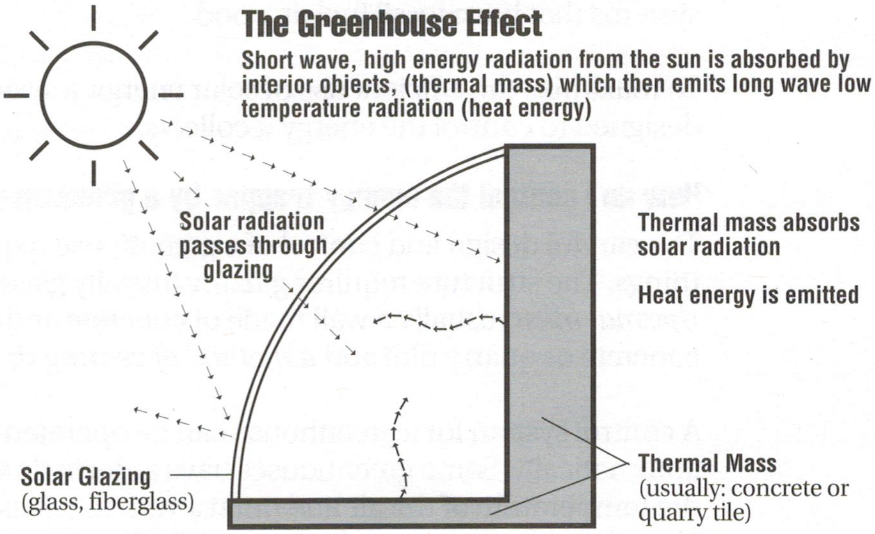
Have you ever opened the door of a car that has been sitting in the sun with its windows rolled up? If you have, you have experienced the greenhouse effect.

The *greenhouse effect* is the ability of transparent materials (such as glass) to allow sunlight into an enclosed area and trap the resulting heat.

A *greenhouse* is a building that is designed to trap the sun's energy. The warm temperature of a greenhouse allows plants to grow throughout the year, even in climates with harsh winters. A *sunspace* operates in the same way a greenhouse does but a sunspace is not designed for plants, it is designed for people.

The Greenhouse Effect

Short wave, high energy radiation from the sun is absorbed by interior objects, (thermal mass) which then emits long wave low temperature radiation (heat energy)


Solar radiation passes through glazing
Thermal mass absorbs solar radiation
Heat energy is emitted
Solar Glazing (glass, fiberglass)
Thermal Mass (usually: concrete or quarry tile)

What is solar energy?

The fundamental source for our energy on Earth is the sun. The effects of solar energy on plant and animal life from long ago is stored as oil, natural gas and coal. *Fossil fuels* is the term used to describe oil, coal and natural gas. Fossil fuels were formed by the tremendous pressure caused by layers of plant and animal remains. These fossil fuels must be extracted from the Earth. They are burned to release the energy they contain.

Wind is another form of energy. It is the result of the sun's warming of the Earth's atmosphere and surface. The warming and cooling of the air causes it to move. Air movement is wind.

Although these forms of energy can be traced back to the sun, the term *solar energy* refers to the direct use of the sun's light or *solar radiation.* Using energy from the sun is better than using fossil fuels for several reasons:

- Solar energy does not require resources to "discover" it underground.
- Solar energy does not require mining or pumping systems to extract the energy.
- Solar energy does not burn and create air pollution.
- The sun provides an almost limitless supply of energy.

Like solar energy, wind does not need to be "discovered," or extracted from the Earth, or burned.

Unfortunately, wind and solar energy do have limitations. What would happen on a still day if your home heating system depended on wind energy? What if your hot water was heated by a solar energy system and the day was cloudy and cold? Most direct solar systems have back up systems that burn fossil fuel or wood.

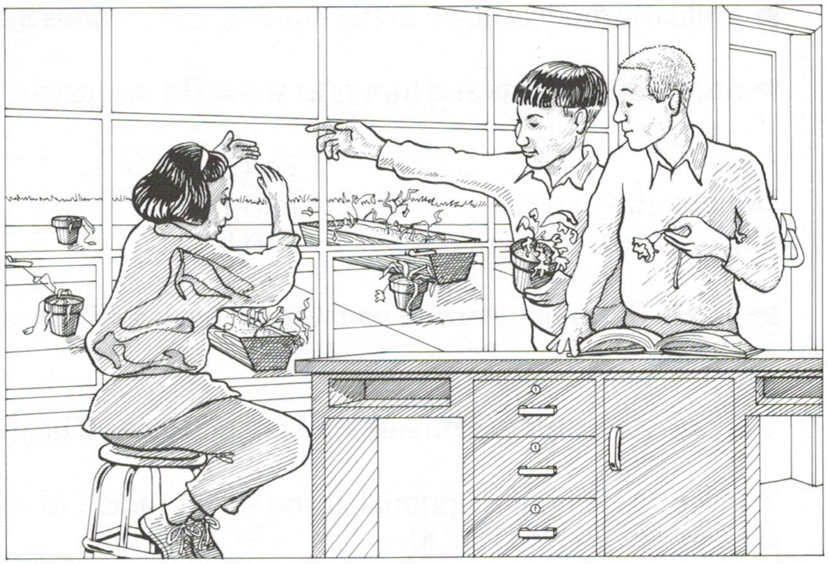
To maximize the efficient use of solar energy, a system must be carefully designed to control the energy it collects.

How do I control the energy trapped by a greenhouse?

The careful design and control of a greenhouse requires three main things. The structure requires *glazing* (usually glass or fiberglass), a *thermal mass* (usually a wall made of concrete and a floor made of concrete or quarry tile) and *a method of venting the air.*

A control system for a greenhouse can be operated manually or automatically. Some greenhouses have automatic systems that monitor the temperature of the air and open a vent to release overheated air. Some greenhouses have shutters or blinds to block strong radiation.

Define the Problem



Problem Statement

The student committee has decided to sell flowers on holidays as a way of raising funds. They need a system designed to grow plants all year around.

Controls for the system must maintain the temperature at an appropriate level for the flowers so they do not die over school vacations.

Design Brief

▶ Choose the type of flowers to grow (e.g.carnations).

▶ Design and build a greenhouse. Program a control system that operates the greenhouse.

The greenhouse design must include:
- Glazing to let in light energy.
- A floor and wall for storing heat energy (thermal mass).
- A method for venting the air.
- A temperature sensor.
- Computer controls that can be used to maintain a correct range of temperature for the flowers.

Additional Criteria

The greenhouse design can also include:
- A feedback system for showing the position of the vent.
- Automatic system controls to check the temperature and raise or lower the window vent.
- A graph that shows the greenhouse temperature and vent position (angle sensor feedback).

Building Ideas

If a sunny window is not available, increase the temperature quickly by placing two to four lamps inside the greenhouse and connecting the lamps to the test port.

To decrease the temperature faster when the window panel is open, swab cold tap water on the end of the temperature sensor with a piece of paper towel ; or attach a temperature sensor from another set to port 1 and dip it in cold tap water.

You must never drip water on any electrical part. Water should only touch the probe of the temperature sensor.

▶ Build the model shown in the building instructions 9701-1.

▶ Attach the outputs and inputs as shown in the instructions.

Motor	Port B
Temperature sensor	Port 1
Angle sensor	Port 6

▶ Attach the angle sensor if you want to have feedback on the vent position.

You can use lamps to simulate sunlight and give off heat energy.

The sound element is optional. If you wish to add it to your control system, attach it to Port A.

Programming Ideas

▶ Start LEGO DACTA® Control Lab software. Create a New Project.

If you wish to measure in degrees Fahrenheit, double-click on the `temp1` icon and select Fahrenheit from the Measure popup menu.

▶ Use the Setup Page to check the motor and temperature sensor.

While testing the temperature sensor and vent motor, you must decide the following:

- What type of flowers am I planting?
- What is the appropriate maximum and minimum temperature range for these flowers?
- At what temperature should the window vent open?
- At what temperature should the window vent close?

Refer to the *Quick Start Guide Appendix A* "Using the Angle Sensor" for more information.

If you are using the angle sensor, you must find out:

- At which rotational count is the vent open "far enough."
- At which rotational count is the vent closed "enough."

▶ Create a Control Panel to operate the greenhouse. Review the design brief specifications so that you include the important control features.

Refer to the *Quick Start Guide* section "Using the Graph Tool" for more information.

▶ If you are graphing the temperature and angle sensor feedback, add the graph to a project page.

▶ Then figure out the maximum and minimum y axis values to use for the graph.

Teacher Notes for Greenhouse: A Bio-related Technology Investigation

✏ Teacher note:

See the "What Do Students Learn?" section at the beginning of this book for concepts in technology, science, math, language arts and social studies that are addressed by the Greenhouse activity.

Objectives

The purpose of the Greenhouse investigation is to introduce students to bio-related technology systems through building and programming a greenhouse. By the end of the project, students will be able to:

1. Describe the process they used to develop their greenhouse.
2. Explain how solar energy can be collected and used.

Students will also develop specific skills working with the LEGO DACTA® Control Lab output and input devices and control features. By the end of the project, students will be able to:

1. Control the movement of one power source (motor).
2. Use pulley and gear ratios to reduce the speed of the power source.
3. Use software tools and commands to develop a control panel for the greenhouse.
4. Use feedback from sensors to control the greenhouse using the Control Lab software.

Materials Needed

- One LEGO DACTA Technology Set and building instruction 9701-1 for each group (2 - 3 students per group recommended).
- At least one Control Lab system.
- Photocopies of the student Greenhouse pages.
- Disks or disk storage space for each student group project.
- Cups of hot and cold tap water for each computer system used (optional).

Background Information

To prepare students, two pages of background information provide answers to the following questions: What is bio-related technology? What is a greenhouse? What is solar energy? How do I control the energy trapped by a greenhouse?

See the Bibliography in the Supplemental Materials section of this book for more information on bio-related technology and other approaches to this area of technology education.

Passive solar energy techniques are discussed in this activity. Active solar energy techniques are not.

Define the Problem

Several criteria are listed on the student design brief page. You may wish to add your own.

If you want students to use angle sensor feedback in their greenhouse control system, they may need to review *Quick Start Guide Appendix A* "Using Angle Sensor" prior to programming the greenhouse.

The examples that follow use the temperature range for carnations for programming the greenhouse. Carnations germinate best at a temperature range of 21-24 °C or 70-75 °F. Students may wish to design their greenhouse around a different type of plant and temperature range.

✏ **Teacher note:**

The Ball Seed representative says that if you call and request a catalog and you identify yourself as an educator, you may be able to receive an old catalog free.

Add books on plants, greenhouse design and solar energy to the classroom resource area. One good source is seed catalog companies. They sell catalogs with temperature ranges for their seeds. One such company is Ball Seed, 622 Town Road, West Chicago, IL 60185-2698. Telephone (800) TRY-BALL.

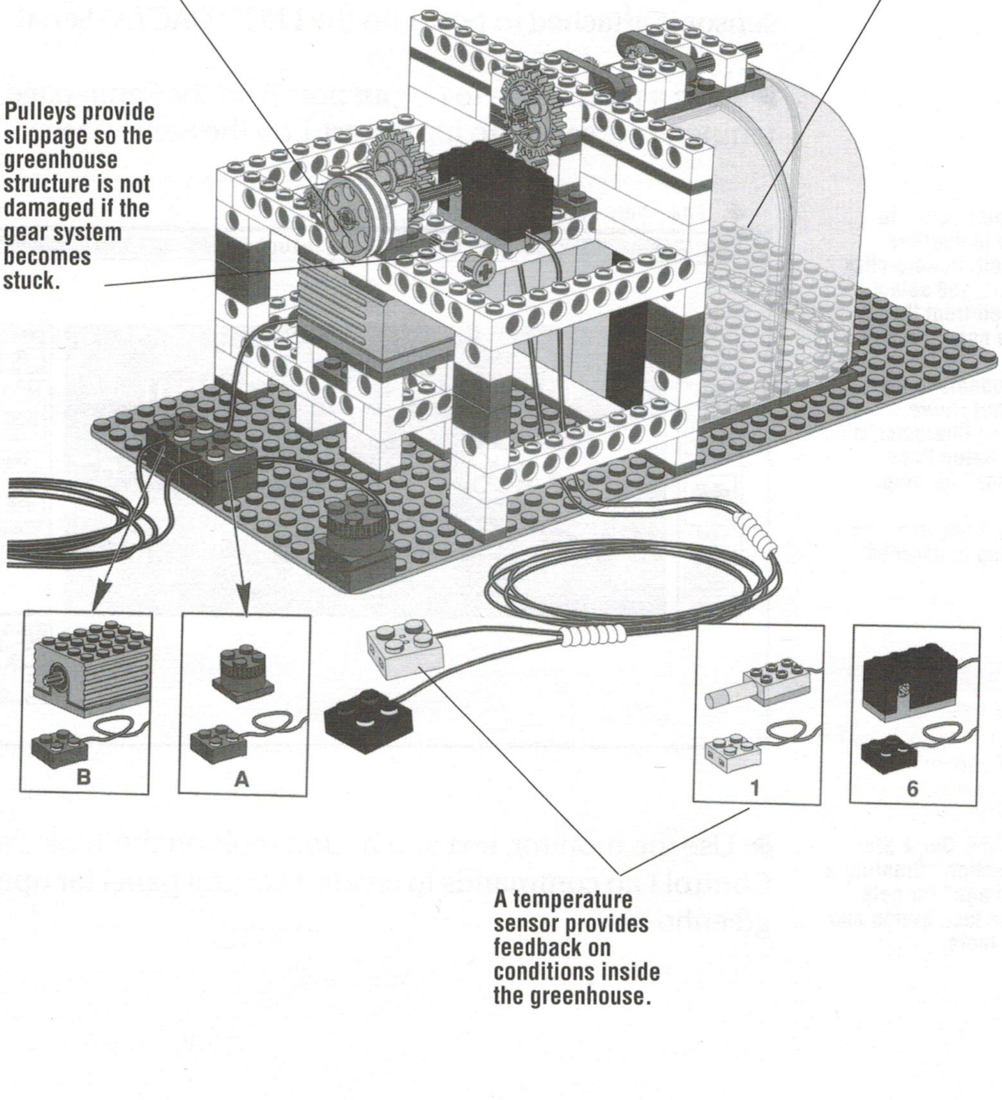
Longwood Gardens is a private horticultural display garden which uses a computer-controlled system designed by QCOM to sense temperature and control vents in their buildings. For more information contact:
Bill Thomas, Education Department, Longwood Gardens, Inc.,
P.O. Box 501, Kennett Square, PA 19348. Telephone (215) 388-6741.
QCOM Corporation, 17782 Cowan Avenue, Irvine, CA 92714.
Telephone (714) 833-1000.

Building Ideas

Students must never drip water on any electrical component. Water should only touch the probe of the temperature sensor.

Students can follow the step-by-step building instructions labelled 9701-1 to create this greenhouse model or they can design and build their own.

The sound element attached to port A is optional. It is not used in the example solutions.



A worm gear slows the movement of the motor so the window opens and slows at a safe rate.

Students can include their own plants!

Pulleys provide slippage so the greenhouse structure is not damaged if the gear system becomes stuck.

A temperature sensor provides feedback on conditions inside the greenhouse.

Programming Ideas Example A: Operator-controlled Greenhouse Vent

Students can use other minimum and maximum temperatures in their control system.

This example addresses the main criteria in the design brief. It shows a greenhouse control system that is not automatic: it is run by the operator. The operator checks the temperature monitor box. He or she is responsible for opening the vent if the temperature rises above 24° C and for lowering the vent if the temperature drops below 21° C.

▶ Start the LEGO DACTA® Control Lab software. Create a New Project.

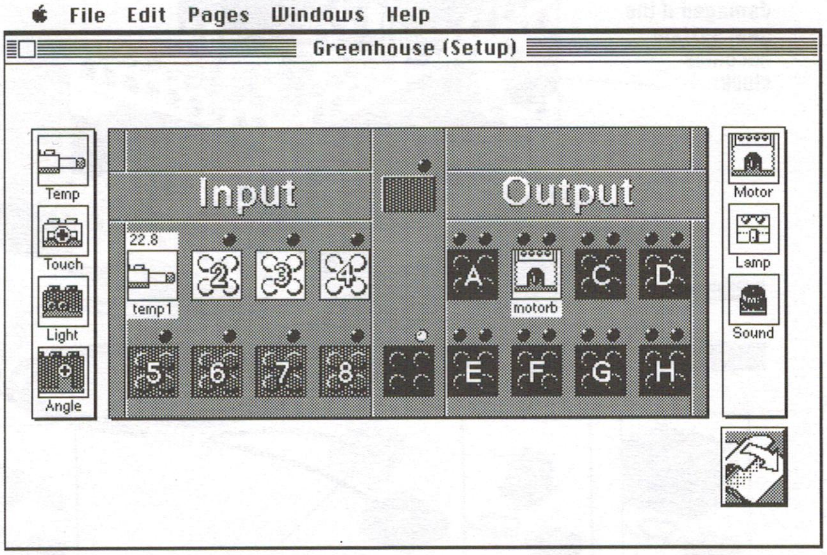
▶ Make sure the vent motor is attached to port B and the temperature sensor is attached to port 1 on the LEGO DACTA® Serial Interface.

▶ Drag a motor icon to output port B on the Setup page. Drag a temperature sensor to input port 1 on the Setup page.

To change `temp1` to measure in degrees Fahrenheit, double-click on `temp1` and select Fahrenheit from the Measure pop-up menu.

Refer students to the *Quick Start Guide* "Changing Characteristics with the Setup Page Dialog Box" for help.

Example A shows `temp1` measuring in degrees Celsius.



Refer to the *Quick Start Guide* section "Creating A Project Page" for help using the text, button and monitor tools.

▶ Use the monitor, text and button tools on the Tools Palette and Control Lab commands to create a control panel for operating the greenhouse.

A heading and an explanatory text box are added to the control panel to make the system easier to understand and use.

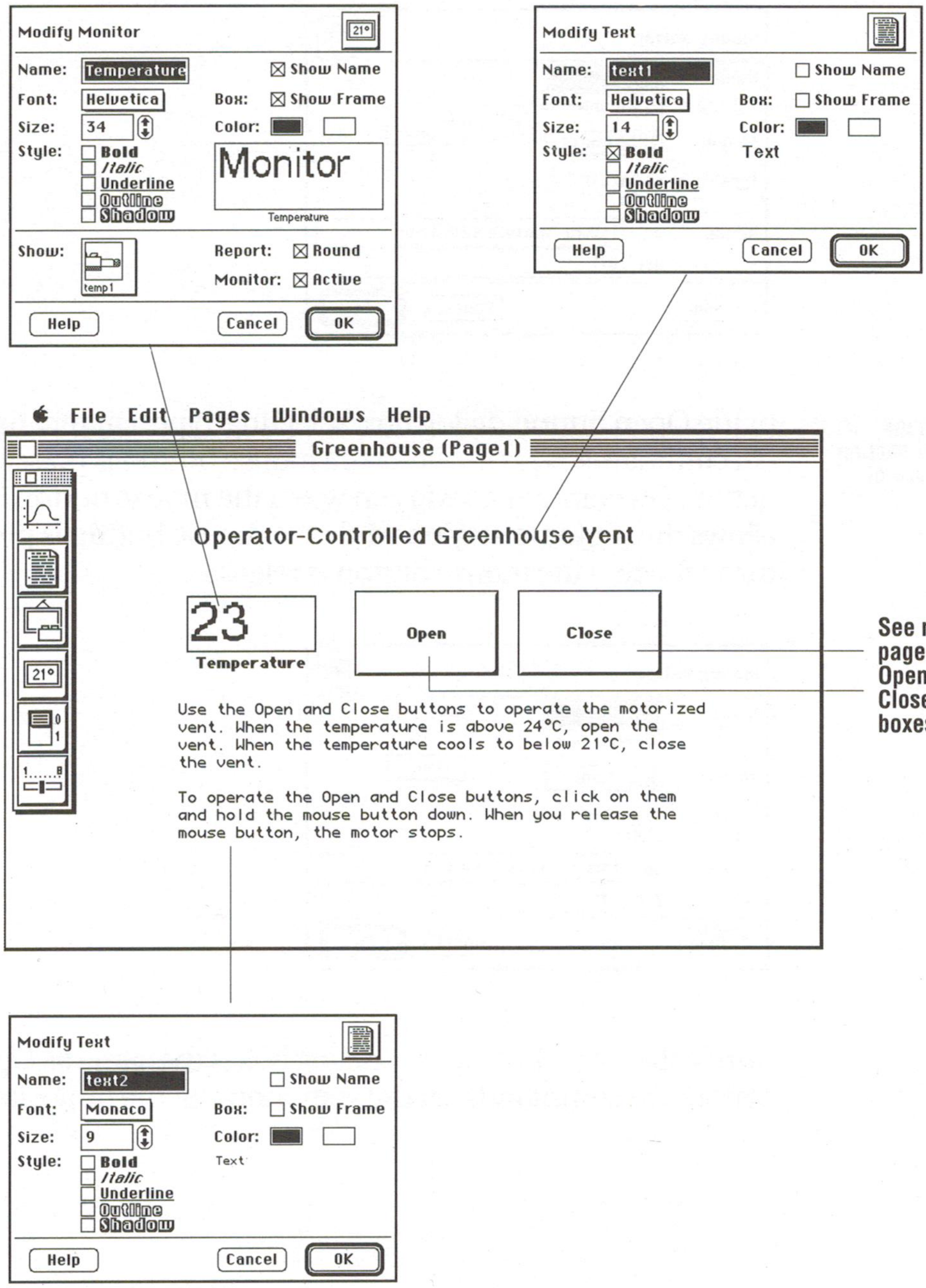
Refer to the *Quick Start Guide* section "Using the Text Tool" for more information.

MS-DOS users:

Text boxes do not include font, size and style.

Macintosh users:

Font, size and style choices depend upon what you have loaded into your computer.



Modify Monitor

Name: Temperature ☒ Show Name
Font: Helvetica Box: ☒ Show Frame
Size: 34 Color:
Style: ☐ Bold ☐ *Italic* ☐ Underline ☐ Outline ☐ Shadow
Monitor
Temperature
Show: temp1
Report: ☒ Round
Monitor: ☒ Active

Modify Text

Name: text1 ☐ Show Name
Font: Helvetica Box: ☐ Show Frame
Size: 14 Color:
Style: ☒ **Bold** ☐ *Italic* ☐ Underline ☐ Outline ☐ Shadow
Text

Operator-Controlled Greenhouse Vent

23
Temperature

Open

Close

Use the Open and Close buttons to operate the motorized vent. When the temperature is above 24°C, open the vent. When the temperature cools to below 21°C, close the vent.

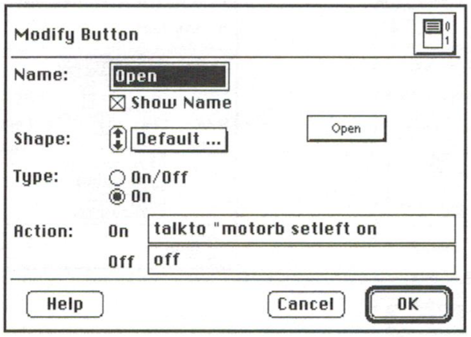
To operate the Open and Close buttons, click on them and hold the mouse button down. When you release the mouse button, the motor stops.

Modify Text

Name: text2 ☐ Show Name
Font: Monaco Box: ☐ Show Frame
Size: 9 Color:
Style: ☐ Bold ☐ *Italic* ☐ Underline ☐ Outline ☐ Shadow
Text

See next page for Open and Close dialog boxes.

To open the vent, the operator holds down the Open button. The motor stays on while the Open button is activated. The motor shuts off when the Open button is released. Likewise, to close the vent, the operator holds down the Close button. No feedback is used from the angle sensor to indicate when the vent is open or closed.



Modify Button

Name: Open

☒ Show Name

Shape: Default ...

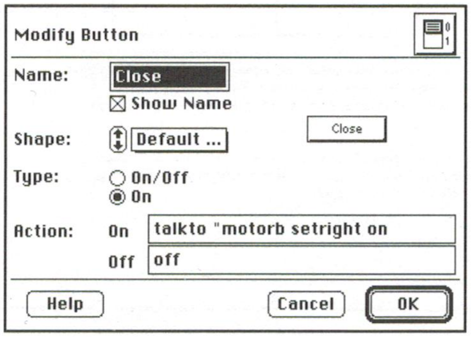
Type: ○ On/Off ● On

Action: On talkto "motorb setleft on

Off off

Refer students to the *Quick Start Guide* section “Changing the Type of Button” for help.

In the Open button dialog box, selecting Type: On means that the Action: On commands will run while the mouse button is pressed and the Action: Off commands will run when the mouse button is released. This allows the motor to stay on while the mouse button is pressed and to turn off when the mouse button is released.



Modify Button

Name: Close

☒ Show Name

Shape: Default ...

Type: ○ On/Off ● On

Action: On talkto "motorb setright on

Off off

Notice that the Close button dialog box is the same as Open, except the Action: On commands set the vent motor in the opposite direction.

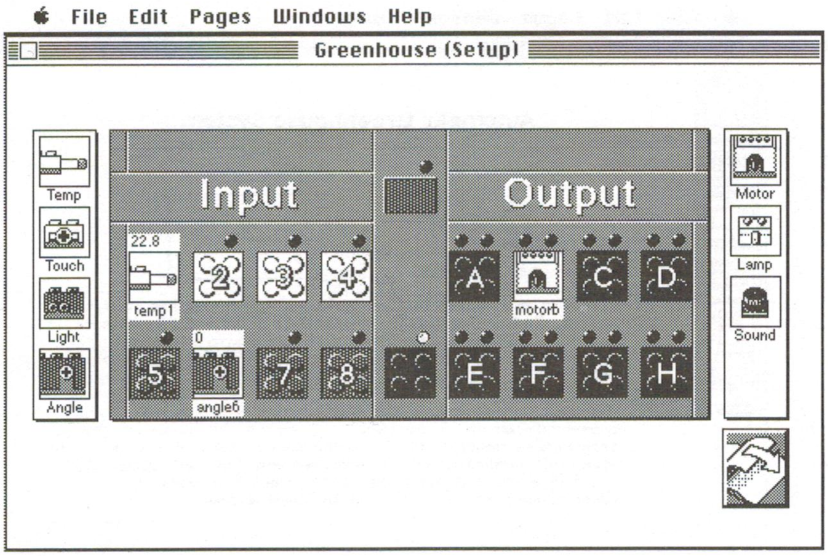
Programming Ideas Example B: Automatic Temperature and Vent Position Control

This example addresses the main criteria and the additional criteria in the design brief. It shows how to use the angle sensor for feedback on the vent position; how to create an automatic system that checks the temperature and raises or lowers the vent; and how to graph the greenhouse temperature and vent position.

▶ Load the LEGO DACTA® Control Lab software. Start a New Project or continue with an open greenhouse project.

▶ Make sure the vent motor is attached to port B, the temperature sensor is attached to port 1, and the angle sensor is attached to port 6 on the LEGO DACTA® Serial Interface.

▶ Drag a motor icon to output port B on the Setup page. Drag a temperature sensor to input port 1 and an angle sensor to input port 6 on the interface box.



To change `temp1` to measure in degrees Fahrenheit, double-click on `temp1` and select Fahrenheit from the Measure pop-up menu.

Refer students to the *Quick Start Guide* "Changing Characteristics with the Setup Page Dialog Box" for help.

Example B shows `temp1` measuring in degrees Celsius.

▶ Click on the circles above the `motorb` icon to open the vent as far as you wish. Write down the angle sensor value.

This is the number to use when programming the greenhouse to open.

Example B uses the number 14.

▶ Click on the circles above the `motorb` icon to close the vent. Write down the angle sensor value.

This is the number to use when programming the greenhouse to close.

Example B uses the number 0.

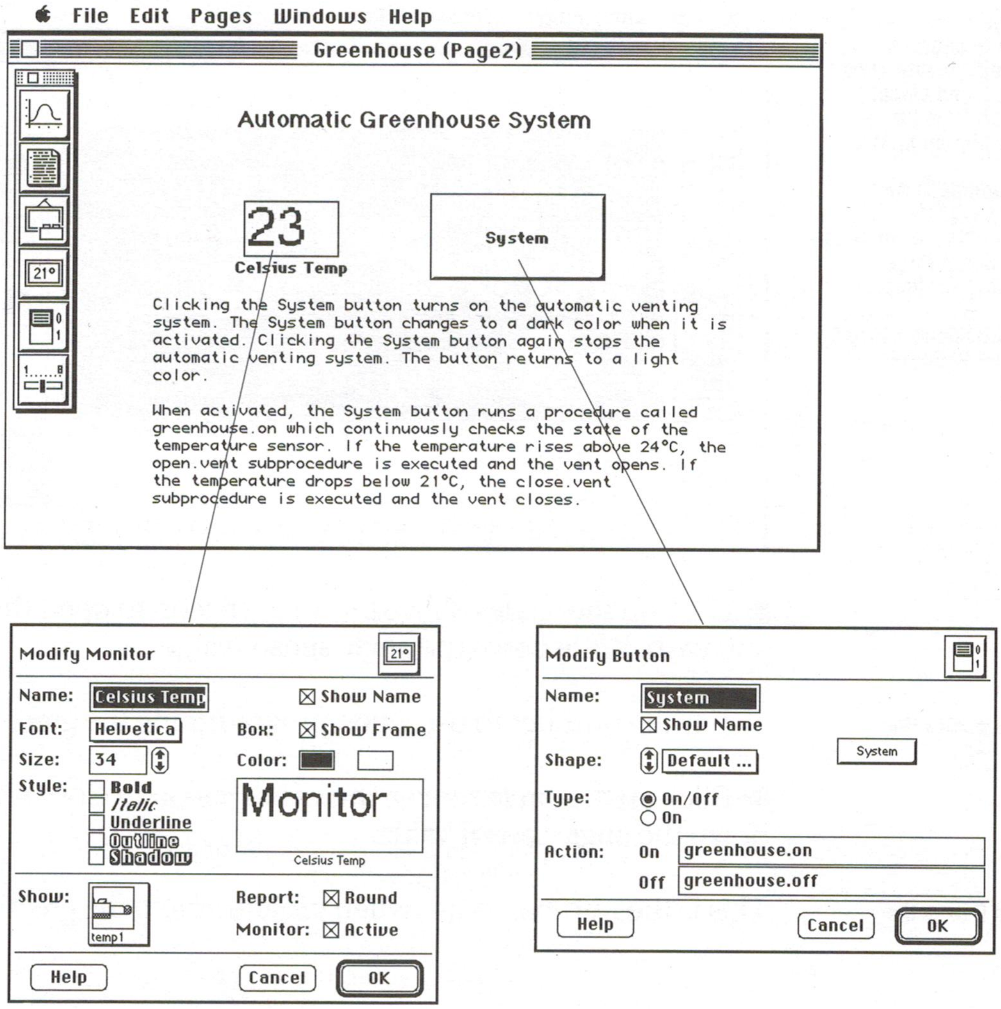
The angle sensor values you choose to use may vary from those shown in the example.

▶ Create a control panel to operate the greenhouse using angle sensor feedback.

Example B below includes the following features:

- A monitor box named Celsius Temp that displays the temperature inside the greenhouse.
- A button that starts and stops the `greenhouse.on` procedure on Procedures page.
- Procedures and subprocedures on the Procedures page that check the temperature and open and close the vent accordingly.
- Text boxes for the heading and explanatory information.
- A graph on a separate page that tracks the temperature and angle sensor feedback.

The heading and explanatory text boxes are created like those shown in Example A. Refer to Example A or the *Quick Start Guide* sections "Using the Text Tool" and "Adding Text to Document Your Work."



Automatic Greenhouse System

Celsius Temp

Clicking the System button turns on the automatic venting system. The System button changes to a dark color when it is activated. Clicking the System button again stops the automatic venting system. The button returns to a light color.

When activated, the System button runs a procedure called greenhouse.on which continuously checks the state of the temperature sensor. If the temperature rises above 24°C, the open.vent subprocedure is executed and the vent opens. If the temperature drops below 21°C, the close.vent subprocedure is executed and the vent closes.

Modify Monitor

Name: Celsius Temp ☒ Show Name

Font: Helvetica Box: ☒ Show Frame

Size: 34 Color:

Style: ☐ Bold ☐ Italic ☐ Underline ☐ Outline ☐ Shadow

Monitor

Celsius Temp

Show: temp1

Report: ☒ Round

Monitor: ☒ Active

Modify Button

Name: System

☒ Show Name

Shape: Default ...

System

Type: ● On/Off ○ On

Action: On greenhouse.on

Off greenhouse.off

The procedure name used in a dialog box must match the procedure name on the Procedures page exactly, or Control Lab responds with an error message.

Notice in the dialog box for the System button, that procedure names are typed in the Action: On and Action: Off fields. That means the procedure is executed when the button is activated. All procedures are typed and stored on the Procedures page. LEGO DACTA® Control Lab software finds the procedure on the Procedures page and executes it.

Writing procedures on the Procedures page makes writing and editing Logo statements much easier, especially if they are more than two or three commands long.

▶ Click the System button to run the automatic control system.

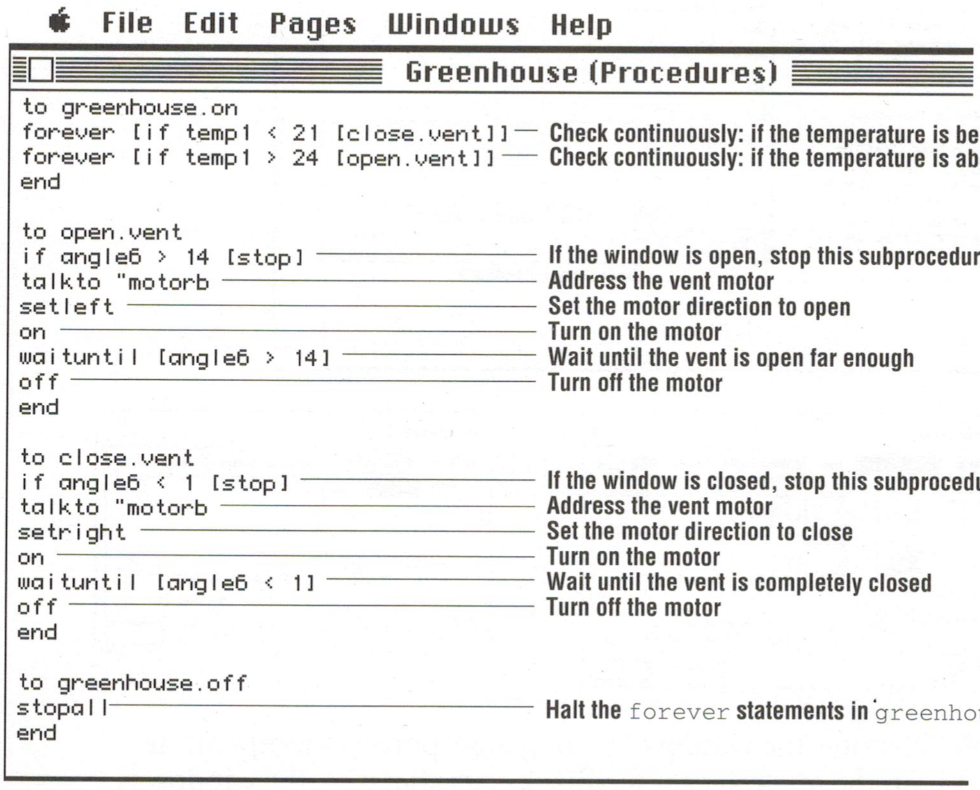
The main procedure (or superprocedure) is called `greenhouse.on`.

`Greenhouse.on` constantly checks the temperature sensor. If the air temperature is too hot, `greenhouse.on` runs the `open.vent` subprocedure on the Procedures page. If the air temperature is too cold, `greenhouse.on` runs the `close.vent` subprocedure on the Procedures page.

Students can review examples of `forever` in the *Quick Start Guide* sections "Using the Touch Sensor to Control the Motor" and "Controlling the Fan Motor with the Temperature Sensor."

Once activated, the `forever` statements in the `greenhouse.on` procedure run continuously.

▶ Click the System button again to run the `greenhouse.off` procedure which halts the `forever` statements in the `greenhouse.on` procedure.



```
to greenhouse.on
forever [if temp1 < 21 [close.vent]]
forever [if temp1 > 24 [open.vent]]
end

to open.vent
if angle6 > 14 [stop]
talkto "motorb
setleft
on
waituntil [angle6 > 14]
off
end

to close.vent
if angle6 < 1 [stop]
talkto "motorb
setright
on
waituntil [angle6 < 1]
off
end

to greenhouse.off
stopall
end
```

Check continuously: if the temperature is below 21°C, close the vent
Check continuously: if the temperature is above 24°C, open the vent

If the window is open, stop this subprocedure and return to `greenhouse.on`
Address the vent motor
Set the motor direction to open
Turn on the motor
Wait until the vent is open far enough
Turn off the motor

If the window is closed, stop this subprocedure and return to `greenhouse.on`
Address the vent motor
Set the motor direction to close
Turn on the motor
Wait until the vent is completely closed
Turn off the motor

Halt the `forever` **statements in** `greenhouse.on`

The graph is created on a new page of the Greenhouse project so it can have more space.

Graphing the temperature and angle sensor feedback is an additional criterion on the student design brief.

▶ Select New Page from the Pages menu.

▶ Select the graph tool and click in an open area of the page to create a graph box.

▶ Double-click the graph box to open the dialog box.

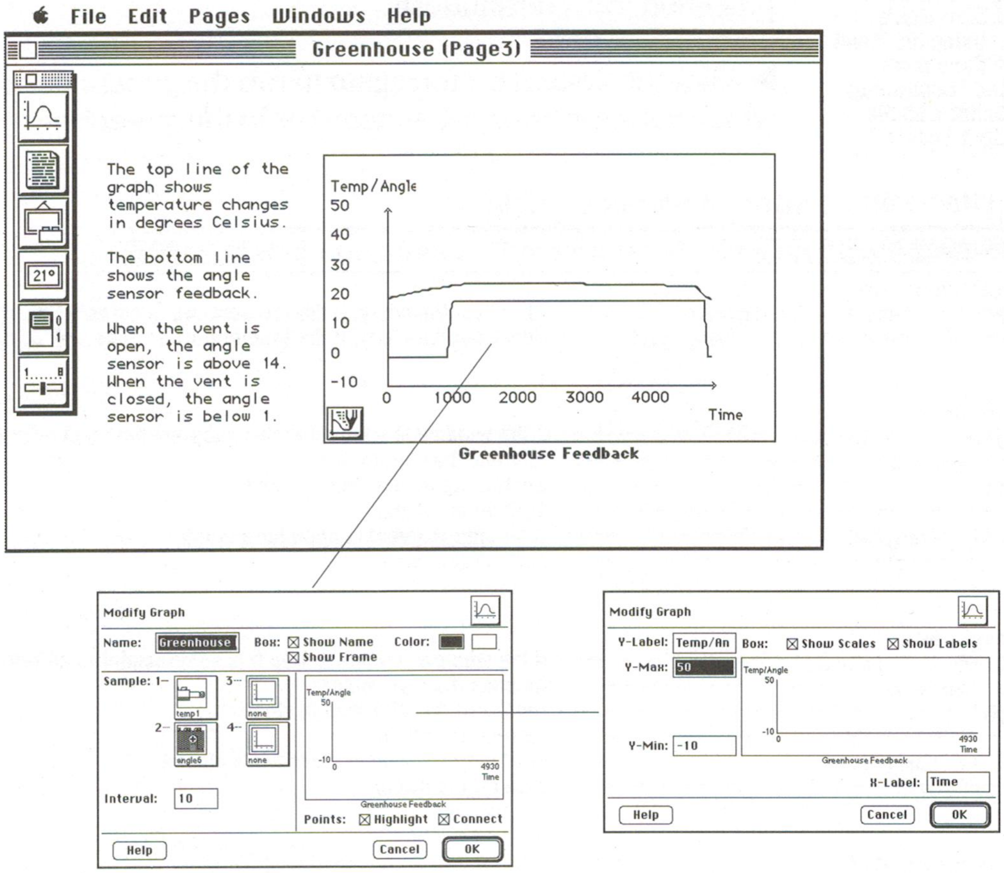
▶ Press down and hold on the popup menu for Sample1. Select `temp1`.

▶ Press down and hold on the popup menu for Sample2. Select `angle6`.

▶ Click on the graph area in the dialog box to open the second dialog box.

Example A uses degrees Celsius and the angle sensor range between 0 and 15, so Y-max is set to 50 and Y-min is set to -10.

▶ Change the Y-max and Y-min to suit the Celsius or Fahrenheit temperature range and the angle sensor feedback.



The top line of the graph shows temperature changes in degrees Celsius.

The bottom line shows the angle sensor feedback.

When the vent is open, the angle sensor is above 14. When the vent is closed, the angle sensor is below 1.

Name: Greenhouse

Sample: 1– temp1 2– angle6 3– none 4– none

Interval: 10

Y-Label: Temp/An

Y-Max: 50

Y-Min: -10

X-Label: Time

The graph illustrates the window opening in response to temperature changes. When the angle sensor feedback goes above 14, the window is open. When the angle sensor drops below 1, the window is closed.

Related Concepts and Extensions (Interdisciplinary)

Technology

- Add a fan to the greenhouse to circulate the air and cool the greenhouse faster.

- Use a touch sensor instead of the angle sensor to indicate when the vent is open or closed.

- Belt drives are included in the drive train of the greenhouse. Why is it important to include belts? *For safety reasons it is important to include belts. Belts allow the slippage needed to prevent the window from closing too hard and damaging the greenhouse structure and drive train.*

Science

- Place a temperature sensor on the floor of the greenhouse and attach it to input port 2. Does the temperature sensor on the floor measure a different amount than the temperature sensor nearer the ceiling? *Yes. Warm air rises, so a temperature sensor mounted on the floor of the greenhouse often measures a lower temperature than a temperature sensor mounted nearer the ceiling.*

- If a window or grow light is available in the classroom, students can use the greenhouse system to experiment with the growing alfalfa or bean sprouts in a container under different lighting and temperature conditions. Light intensity can be monitored with the light sensor.

✏ Teacher note:

Students should check that both sensors are reading the same temperature by placing both sensors in a glass of water first. They can adjust the sensor readings in the dialog boxes on the Setup Page.

Students must remember to remove the container when watering it so they do not get water on the electrical components of Control Lab.

Math

- Add a light sensor into the greenhouse. Graph the light sensor and temperature changes. Do the graphs show that temperature increases when light increases? *Yes, however, temperature changes take longer.*

- To convert degrees Celsius to degrees Fahrenheit, multiply by 1.8, then add 32. To convert Fahrenheit to Celsius, subtract 32, then divide by 1.8.

- Calculate the gear and pulley ratios of the greenhouse window raising mechanism. Gear teeth and pulley diameter can be used in the following formula.

$$\text{RATIO} = \frac{\text{OUTPUT}}{\text{INPUT}} \text{ X } \frac{\text{OUTPUT}}{\text{INPUT}} \text{ X } \frac{\text{OUTPUT}}{\text{INPUT}} \text{ X ETC.}$$

Approximate gear ratios for the greenhouse model pictured in the Building Ideas are as follows (for the pulleys compare the diameters):

$$\frac{3}{1} \text{ x } \frac{24}{1} \text{ x } \frac{3}{1} \text{ x } \frac{2}{8} = \frac{648}{1}$$

Or more precisely (diameter of pulleys in inches):

$$\frac{.875}{.250} \text{ x } \frac{24}{1} \text{ x } \frac{.850}{.250} \text{ x } \frac{24}{8} = \frac{3.5}{1} \text{ x } \frac{24}{1} \text{ x } \frac{3.5}{1} \text{ x } \frac{3}{1} = \frac{882}{1}$$

For centimeters use 2.222 / 0.635 instead of .875 / .250. For some math conversion practice, you could tell students to multiply the inches numbers by 2.54 to find the result.

Ratios can be expressed as fractions or as INPUT : OUTPUT. The latter is found more often in technical manuals.

Language Arts

- Design and write an advertisement to post throughout the school promoting the purchase of the flowers to help fund a school event.
- Ask students to write a proposal to the school administration, listing the costs of the greenhouse and explaining the work to be done.

Social Studies

- You are in charge of fundraising for the student committee. Make a list of the flowers and plants you recommend selling at different holidays throughout the school year. List the type of flower or plant, dates you plan to sell them and the temperature range at which they must be kept in order to stay healthy.
- Visit a flower or garden shop and interview the owner or employee. How do they organize the plants so people can find them? What types of plants do they order and when do they order them? Are there some plants that are sold all year around? Why? What plants are used in the holiday seasons?

Robot: A Production / Manufacturing Investigation

Student Name: ______________________________ **Date:** ______________

Background Information

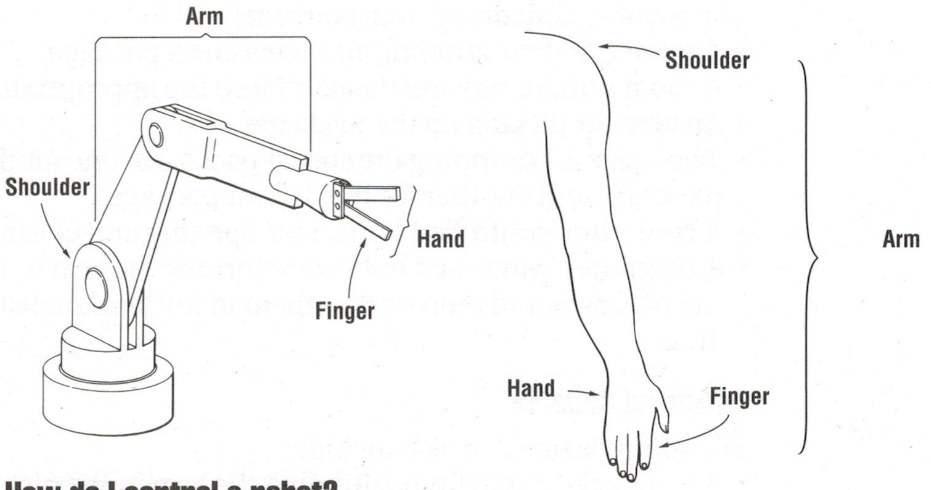
What is Production / Manufacturing technology?

Production involves changing resources into materials that can be purchased by consumers. For example, raw materials such as crude oil can be extracted and refined into other usable materials such as oil and gasoline. Manufacturing is a specific type of production. During the manufacturing process, workers make products at a specific location. For example, automobiles are manufactured at an assembly plant.

What is a robot?

A *robot* is an automatic, computer-controlled machine. Robots can be found in many places. For example, the lunar and Viking landers collected data from the moon and Mars. In some medical facilities robots are used to handle dangerous materials. The entertainment industry has developed robots that interact with people for fun.

One important type of robot is the industrial robot. The movement of the industrial robot imitates those of a human arm. In fact, the words used to describe parts of the robot are like those of the human body. The gripper at the end of the robot is called the *hand.* The part of the robot manipulating the hand is the *arm.* The base which supports the arm and hand is called the *shoulder.*

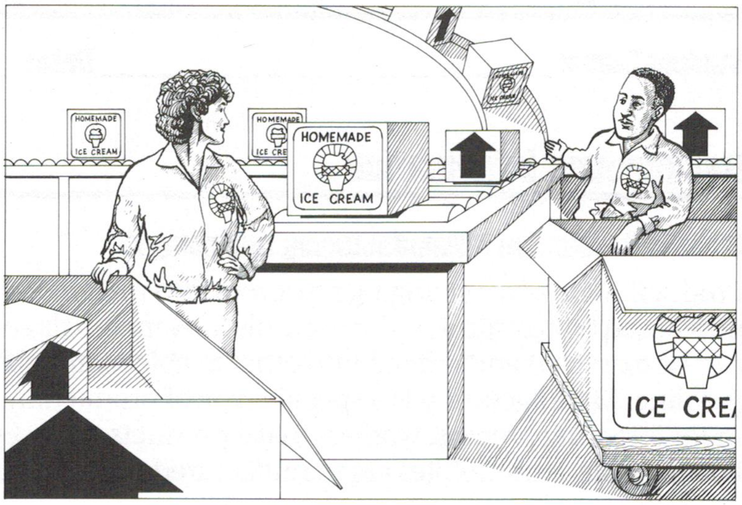


Drawing concept revised from one shown in the book, Fuller, James L. *Robotics: Introduction, Programming, and Projects*. MacMillen Publishing Co. New York: NY, 1991 page 40.

How do I control a robot?

Controlling a robot involves programming a sequence of steps that can be repeated. Sensors can be used to provide feedback on the robot's position so the steps are repeated reliably and safely.

Define The Problem



Problem Statement

Homemade Ice Cream Company packages ice cream for delivery to local stores and for shipment to distant stores. Local packages have an ice cream decal on them. Distant packages have a black "shipment" arrow on them. Both types of packages come down the chute at the end of the packing line and must be placed into the appropriate shipping bin. The packages are bulky and difficult to move by hand.

Design Brief

▶ Design and build a robot to pick up each ice cream package and place it into the appropriate shipping bin. Program a control panel that operates the robot sorting system.

The robot system design must include:

- A robot hand for gripping and releasing a package.
- A robot arm for moving the hand into the appropriate position.
- An area for picking up the packages.
- Two areas for dropping the sorted packages: one for the local packages and another for the distant packages.
- A computer-controlled button for opening and closing the hand.
- A computer-controlled button for turning the arm so it can pick up the packages and then release them in the local or distant drop-off area.

Additional Criteria

The robot design can also include:

- A feedback system for positioning the arm in the pick up area and the local or distant drop-off areas.
- An area on the computer control panel for recording the number of local and distant packages that are sorted.
- Picture shapes to illustrate the control panel.

Building Ideas

▶ Build the model shown in building instructions 9701-7.

▶ Attach the outputs and inputs as shown in the instructions.

Robot arm motor	Port A
Robot hand motor	Port B
Angle sensor	Port 5

The angle sensor is an additional part of the design brief. It can be used to provide feedback on the position of the robot arm.

Programming Ideas

▶ Load the LEGO DACTA® Control Lab software. Create a New Project.

▶ Use the Setup Page to test the motors.

▶ While testing the robot, you must find out:
- How much time does the hand take to open and to close?
- How much motor overrun is necessary for the hand to clamp a package securely?

If you are using the angle sensor for feedback, refer to *Quick Start Guide Appendix A* "Using the Angle Sensor" for information.

▶ If you are using the angle sensor, you must find out:
- At which rotational count must the robot stop to pick up a package?
- At which rotational count must the robot stop to drop a local package?
- At which rotational count must the robot stop to drop a distant package?

▶ Create a control panel to operate the robot. Review the design brief specifications so that you include the important control features.

Teacher Notes for Robot: A Production / Manufacturing Investigation

✏ **Teacher note:**
See the "What Do Students Learn?" section at the beginning of this book for concepts in technology, science, math, language arts and social studies that are addressed by the Robot activity.

Objectives

The purpose of the Robot investigation is to introduce students to a manufacturing-related situation through building and programming a robot. By the end of the project, students will be able to:

1. Describe the process they used to develop the robot.
2. Explain how robots can be used in manufacturing.

Students will also develop specific skills working with the LEGO DACTA® Control Lab output and input devices and control features. By the end of the project, students will be able to:

1. Control the movement of two aspects of a robot (grasping and turning).
2. Use pulley and gear ratios to reduce the speed of the motor.
3. Use software tools and commands to develop a control panel for the robot arm.
4. Operate the robot using the Control Lab software and a control panel.

Materials Needed

- One LEGO DACTA Technology Set and building instructions for each group (2 - 3 students)
- At least one Control Lab system
- Photocopies of the student Robot pages
- Disks or disk storage space for each student group project

Background Information

To prepare students for the robot activity, a page of background information provides answers to the following questions: What is Production / Manufacturing technology? What is a robot? How do I control a robot?

If you want students to use angle sensor feedback in their robot control system, they may need to review *Quick Start Guide Appendix A* "Using the Angle Sensor" prior to programming the robot.

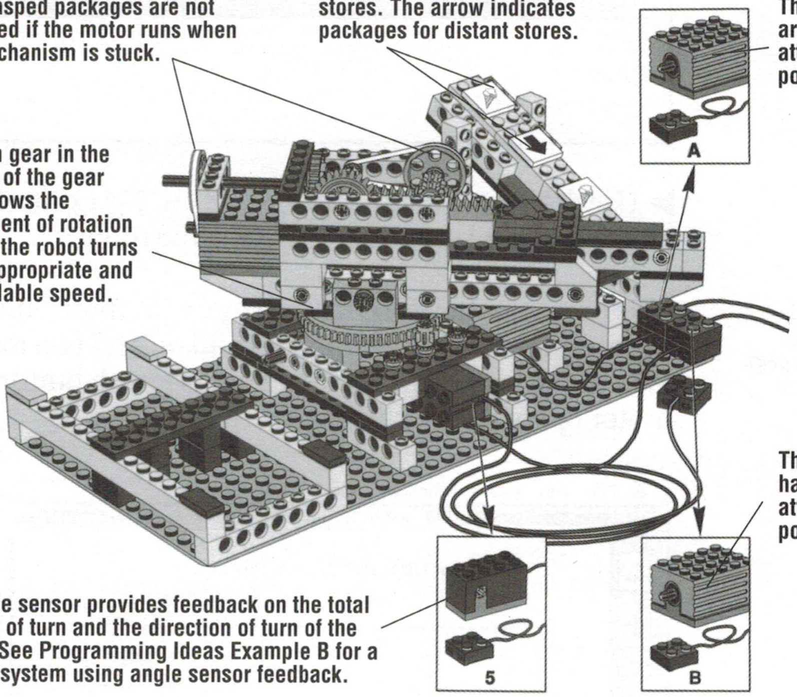
See the Bibliography in the Supplemental Materials section of this book for additional references.

Define The Problem

Several criteria are listed on the student design brief page. You may wish to add your own.

Building Ideas

Students can follow the step-by-step building instructions labelled 9701-7 to create this robot model or they can design and build their own.



Belt drives with pulley wheels provide slippage so that the robot and grasped packages are not damaged if the motor runs when the mechanism is stuck.

The ice cream decal indicates packages for local stores. The arrow indicates packages for distant stores.

The robot arm motor attaches to port A.

A worm gear in the middle of the gear train slows the movement of rotation so that the robot turns at an appropriate and controllable speed.

The robot hand motor attaches to port B.

An angle sensor provides feedback on the total amount of turn and the direction of turn of the robot. See Programming Ideas Example B for a control system using angle sensor feedback.

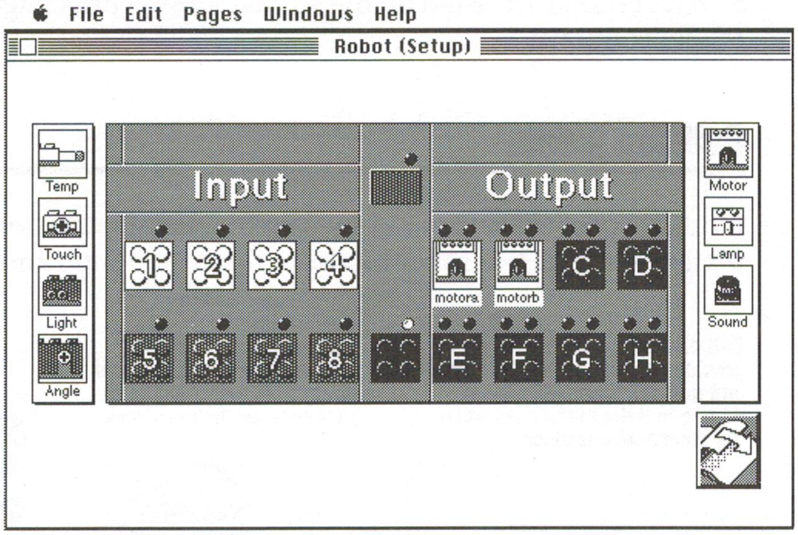
Programming Ideas Example A: Operator-controlled Robot Positioning

▶ Make sure the arm motor is attached to port A and the hand motor is attached to port B.

▶ Load the LEGO DACTA® Control Lab software. Create a New Project.

▶ Drag the motor icons onto output ports A and B on the Setup page.

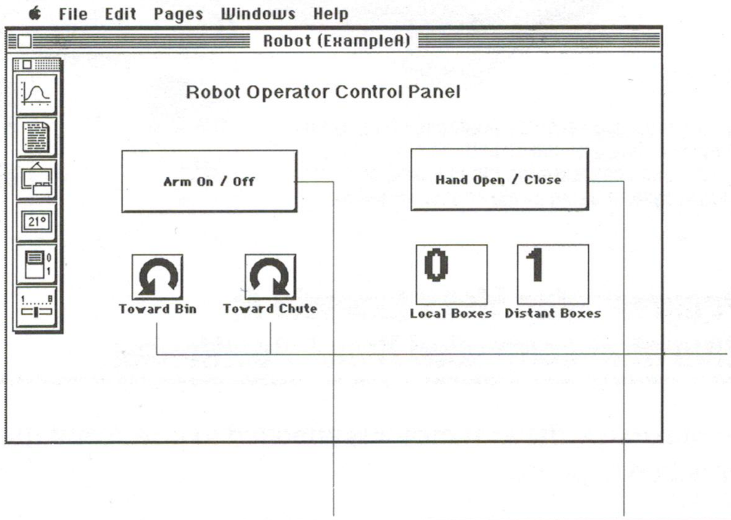
▶ Test the robot arm and hand motors by clicking on the circles above the `motora` and `motorb` icons to make sure they operate properly.



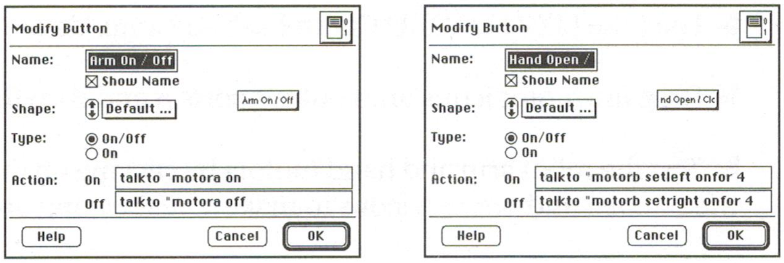
▶ Use the Tools Palette and LEGO DACTA® Control Lab commands to create a control panel for operating the robot.

Students can add their own pictures to the control panel. See page 2.13 for details.

This example shows a control panel with three buttons for controlling the arm motor: one to turn it on and off and two for changing direction toward the bin or toward the chute. A fourth button controls the hand motor opening and closing.



See the Toward Bin and Toward Chute dialog boxes on the next page.



Modify Button

Name: Arm On / Off

☒ Show Name

Shape: Default ...

Type: ● On/Off ○ On

Action: On talkto "motora on

Off talkto "motora off

Modify Button

Name: Hand Open /

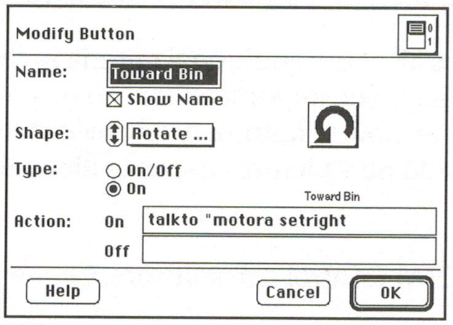
☒ Show Name

Shape: Default ...

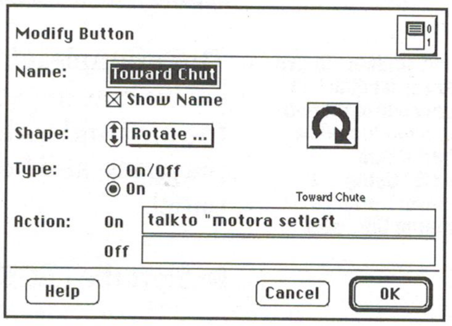
Type: ● On/Off ○ On

Action: On talkto "motorb setleft onfor 4

Off talkto "motorb setright onfor 4



Name: Toward Bin
☒ Show Name
Shape: Rotate ...
Type: ○ On/Off
● On
Action: On talkto "motora setright
Off



Name: Toward Chut
☒ Show Name
Shape: Rotate ...
Type: ○ On/Off
● On
Action: On talkto "motora setleft
Off

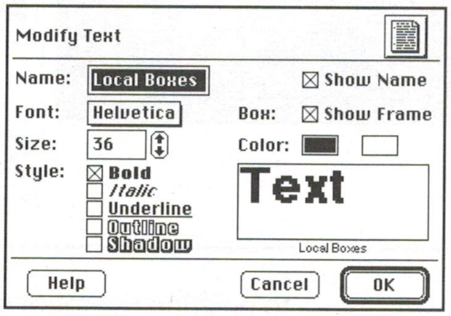
Toward Bin button dialog box and Toward Chute button dialog box.

Additional Criteria: Using a Text Box to Track the Number of Sorted Packages

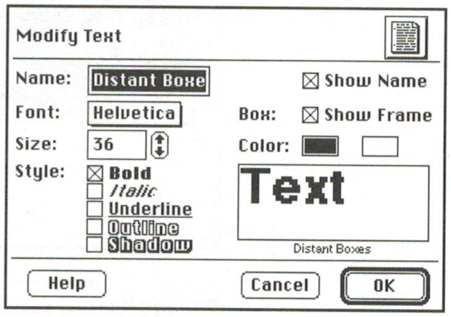
Tracking the number of sorted boxes is an additional criterion in the student design brief. This example shows how to do it manually. Example B shows how to use the monitor tool to track the sorted boxes automatically.

▶ To keep track of the number of local and distant packages that are sorted, create two text boxes: one labelled Local Boxes, the other labelled Distant Boxes.

▶ Then, after each successful sort, type a new number in the appropriate text box.



Name: Local Boxes
☒ Show Name
Font: Helvetica
Box: ☒ Show Frame
Size: 36
Color: ■ ☐
Style: ☒ Bold
☐ Italic
☐ Underline
☐ Outline
☐ Shadow



Name: Distant Boxe
☒ Show Name
Font: Helvetica
Box: ☒ Show Frame
Size: 36
Color: ■ ☐
Style: ☒ Bold
☐ Italic
☐ Underline
☐ Outline
☐ Shadow

MS-DOS users:

Text boxes do not include Font, Size and Style.

Macintosh users:

Font, size and style choices depend on what fonts are loaded into your computer system.

Programming Ideas Example B: Using Angle Sensor Feedback for Position

If you want students to use angle sensor feedback in their robot control system, they may need to review *Quick Start Guide Appendix A* "Using the Angle Sensor" prior to programming the robot arm.

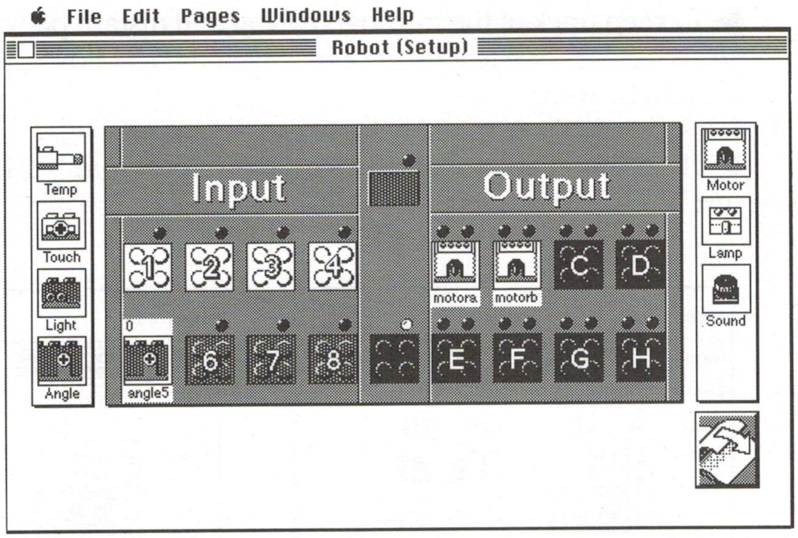
This example addresses all of the additional criteria in the design brief. It shows how to use the angle sensor for feedback to position the robot over the bins and at the chute; how to automatically record the sorted packages; and how to add new picture shapes to illustrate the control panel.

▶ Start the LEGO DACTA® Control Lab software. Create a New Project

Make sure the robot arm motor is attached to output port A and the robot hand motor is attached to output port B. Corresponding icons should be placed on ports A and B of the Setup page.

▶ Attach the angle sensor to input port 5 on the LEGO DACTA Serial Interface.

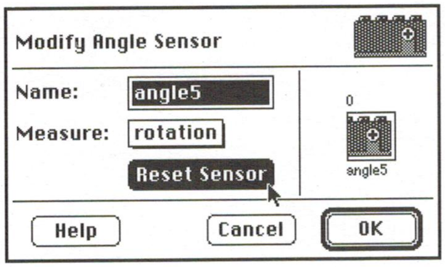
▶ Next, drag an angle sensor icon to port 5 on the Setup page.



▶ Click on a circle above `motora` to position the robot arm at the end of the chute.

▶ Reset `angle5` to zero.

Double-click on `angle5` to open the dialog box. Then click on the Reset Sensor button. Click OK to return to the Setup page. Or, type `resetrotation 5` in the Command Center.


Modify Angle Sensor

Name: angle5

Measure: rotation

▶ Click on one of the circles above the `motora` icon on the Setup page to position the robot over the "distant packages" bin. Write down the value of `angle5`.

This is the number to use when programming the robot to move to the "distant package" bin.

▶ Click on the circles above `motora` icon to position the robot over the "local packages" bin. Write down the angle sensor value.

The readings on the angle sensor may vary from those shown in this example.

This is the number to use when programming the robot to move to the "local package" bin.

Testing the Robot Hand Motor

Use the Setup page to test the robot hand motor which is attached to output port B.

▶ Click on one of the circles above port B to position the hand in the open position over the chute.

The robot should be able to grasp a package as it closes.

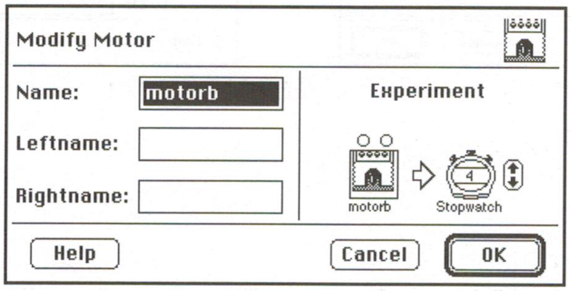
▶ Double-click on the `motorb` icon to open the dialog box for the hand motor.

▶ Click on the circle above the `motorb` icon in the dialog box until the hand motor closes and grasps a "package."

▶ Write down the number in the stopwatch.

This is the amount of time in tenths of seconds the robot hand motor needs to grasp the package. Use this number with the `onfor` command when programming the hand motor to pick up and release packages.

If the stopwatch is not visible, click on the arrow next to the stopwatch until it appears.



Modify Motor

Name: motorb

Leftname:

Rightname:

Creating a Control Panel using Angle Sensor Feedback

▶ Create a control panel to operate the robot using angle sensor feedback.

The example below includes the following features:

- Buttons named Place Local and Place Distant that pick a package from the chute and place it in a bin.
- Monitor boxes named Quantity Local and Quantity Distant that are automatically updated as boxes are added to the bins.
- Buttons named Reset Local and Reset Distant that reset the monitor boxes back to 0.

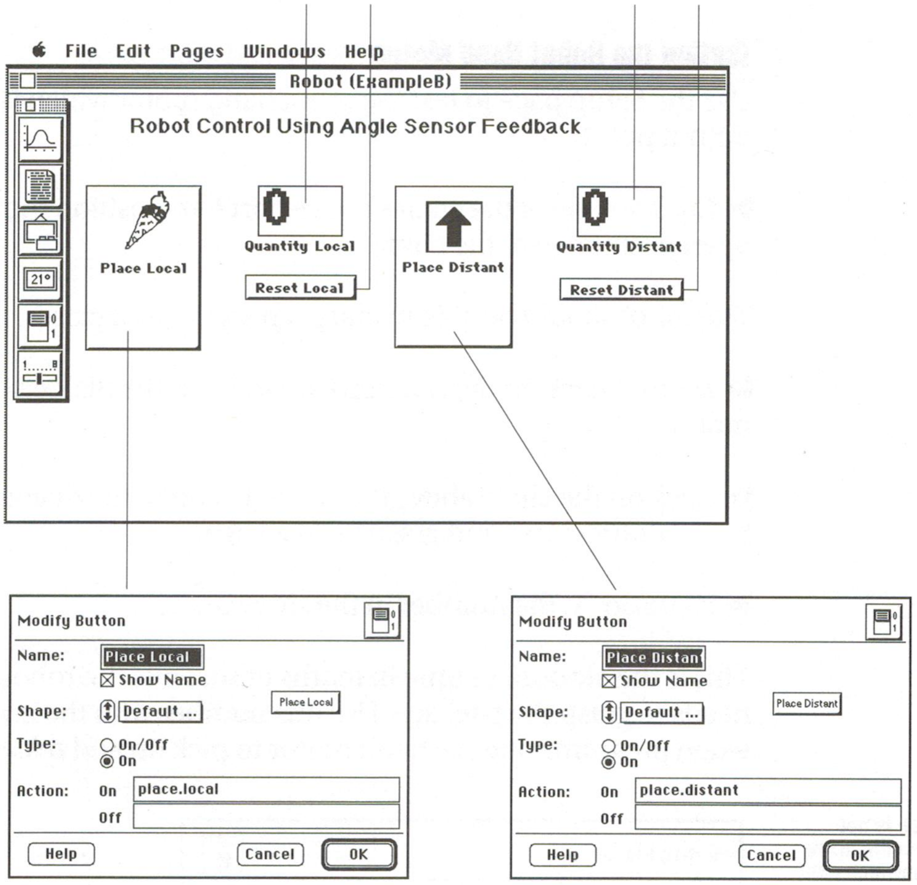
Notice in the dialog boxes for the buttons that procedure names are typed in the Action: On area. That means the procedure is executed when the button is activated.

All procedures are typed and stored on the Procedures page. LEGO DACTA® Control Lab software finds the procedure on the Procedures page and executes it. The procedure name used in a dialog box must match the procedure name on the Procedures page exactly, or Control Lab responds with an error message.

Writing procedures on the Procedures page makes writing and editing Logo statements much easier, especially if they are more than two or three commands long.

See page 2.13 for information about adding user-defined picture shapes to the control panel.

See the dialog boxes for Quantity Local, Reset Local, Quantity Distant and Reset Distant on the next page.



Modify Button

Name: Place Local
☒ Show Name
Shape: Default ...
Type: ○ On/Off ● On
Action: On place.local
Off

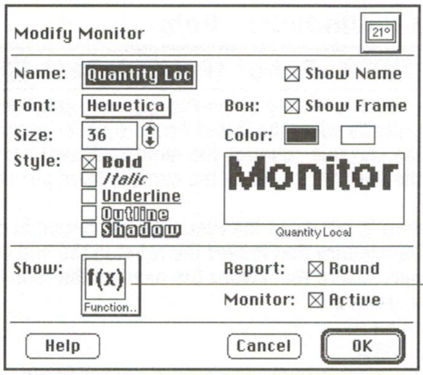
Modify Button

Name: Place Distant
☒ Show Name
Shape: Default ...
Type: ○ On/Off ● On
Action: On place.distant
Off

Students should use their own numbers in the `place.distant` **and** `place.local` **procedures activated by the Place Distant and Place Local buttons. The angle sensor numbers -47 and -57 may not be correct for their model.**

Students should use the Setup page to find out what the correct numbers for their robot system are. See the beginning section in Example B for more information.

Quantity Local monitor dialog box



Modify Monitor

Name: Quantity Loc ☒ Show Name

Font: Helvetica Box: ☒ Show Frame

Size: 36 Color:

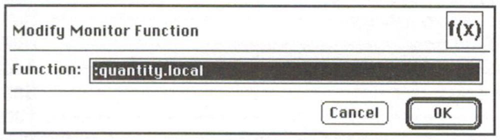
Style: ☒ Bold ☐ Italic ☐ Underline ☐ Outline ☐ Shadow

Show: f(x) Function...

Report: ☒ Round

Monitor: ☒ Active

The monitor will not show a number until the function variable is defined and run by the `startup` procedure (see next page.)

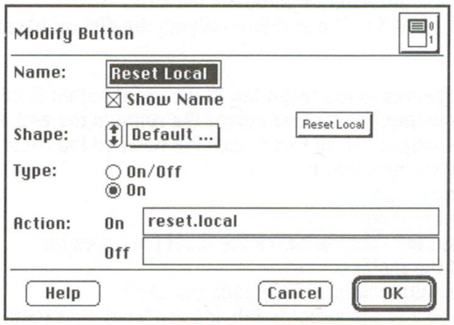


Modify Monitor Function

Function: :quantity.local

Quantity Local monitor function dialog box

Reset Local dialog box



Modify Button

Name: Reset Local

☒ Show Name

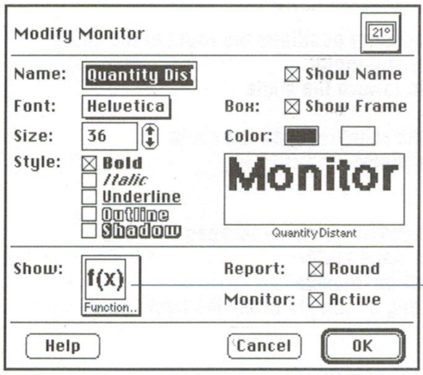
Shape: Default ...

Type: ○ On/Off ● On

Action: On reset.local

Off

Quantity Distant monitor dialog box



Modify Monitor

Name: Quantity Dist ☒ Show Name

Font: Helvetica Box: ☒ Show Frame

Size: 36 Color:

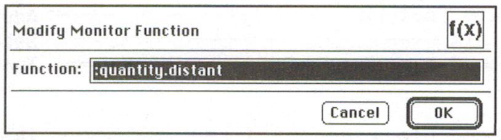
Style: ☒ Bold ☐ Italic ☐ Underline ☐ Outline ☐ Shadow

Show: f(x) Function...

Report: ☒ Round

Monitor: ☒ Active

The monitor will not show a number until the function variable is defined and run by the `startup` procedure (see next page.)

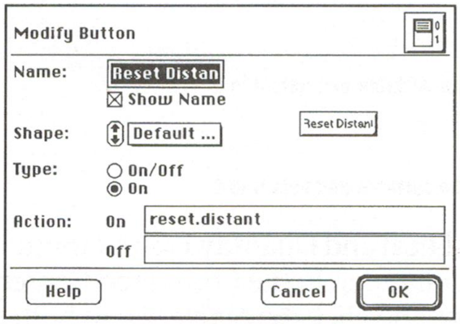


Modify Monitor Function

Function: :quantity.distant

Quantity Distant monitor function dialog box

Reset Distant button dialog box



Modify Button

Name: Reset Distan

☒ Show Name

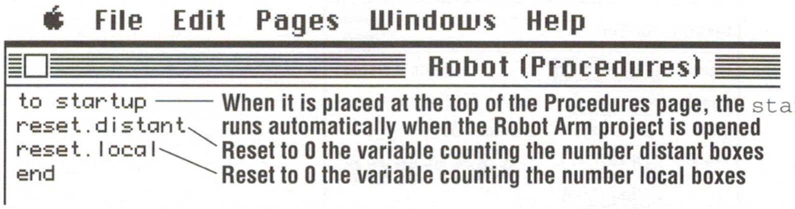
Shape: Default ...

Type: ○ On/Off ● On

Action: On reset.distant

Off

The `startup` procedure must be placed as the first procedure on the Procedures page in order to work correctly.



`to startup`	**When it is placed at the top of the Procedures page, the `startup` procedure runs automatically when the Robot Arm project is opened**
`reset.distant`	**Reset to 0 the variable counting the number distant boxes**
`reset.local`	**Reset to 0 the variable counting the number local boxes**
`end`	

`to place.distant`	**This procedure is run when the Place Distant button is clicked**
`pick.from.chute`	**Run the subprocedure that moves the robot to the end of the chute**
`hand.close`	**Run the subprocedure that closes the hand so the robot picks up a package**
`talkto "motora`	**Address the arm motor**
`setright`	**Set its direction**
`on`	**Turn on the motor**
`waituntil [angle5 < -47]`	**Wait until the robot is above the distant packages bin**
`off`	**Turn off the motor**
`hand.open`	**Run the subprocedure that opens the hand**
`make "quantity.distant :quantity.distant + 1`	**Add 1 to the variable tallying the distant packages**
`end`	

`to place.local`	**This procedure is run when the Place Local button is clicked**
`pick.from.chute`	**Run the subprocedure that moves the robot to the end of the chute**
`hand.close`	**Run the subprocedure that closes the hand so the robot picks up a package**
`talkto "motora`	**Address the arm motor**
`setright`	**Set its direction**
`on`	**Turn on the motor**
`waituntil [angle5 < -57]`	**Wait until the robot is above the local packages bin**
`off`	**Turn off the motor**
`hand.open`	**Run the subprocedure that opens the hand**
`make "quantity.local :quantity.local + 1`	**Add 1 to the variable tallying the local packages**
`end`	

`to pick.from.chute`	**This subprocedure positions the robot at the chute**
`talkto "motora`	**Address the arm motor**
`setleft`	**Set it to turn toward the chute**
`on`	**Turn it on**
`waituntil [angle5 > 0]`	**Wait until the robot reaches the chute**
`off`	**Turn off the motor**
`end`	

`to hand.open`	**This subprocedure grasps packages**
`talkto "motorb`	**Address the hand motor**
`setleft`	**Set its direction to close**
`onfor 4`	**Turn it on long enough to grasp the package**
`end`	

`to hand.close`	**This subprocedure releases packages**
`talkto "motorb`	**Address the hand motor**
`setright`	**Set its direction to open**
`onfor 4`	**Turn it on long enough to release the package**
`end`	

`to reset.local`	
`make "quantity.local 0`	**Creates a variable and sets it to 0**
`end`	

`to reset.distant`	
`make "quantity.distant 0`	**Creates a variable and sets it to 0**
`end`	

To make the Quantity Local and Quantity Distant monitor boxes active (show a number), run `startup` in the Command Center. Then go to the Example B project page and click the Active box in both of the Quantity Local and Quantity Distant monitor dialog boxes.

Additional Criteria: Creating Picture Shapes and Adding Them to the Control Panel

▶ Create ice cream cone and up-arrow drawings in a paint or draw program; or draw them by hand and scan them with a scanner.

Macintosh users: ▶ Save the pictures in a PICT format. Copy a picture into the computer clipboard memory (⌘ - C).

MS-DOS users: ▶ Save the pictures you create in a PCX format.

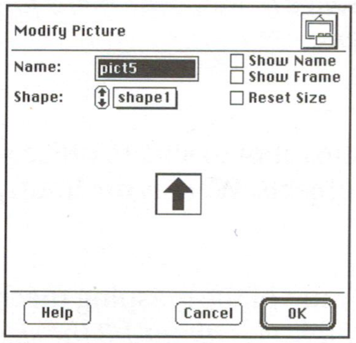
▶ Load LEGO DACTA® Control Lab software. Create a New Project or Open an existing Control Lab project. Go to a project page.

Macintosh users: ▶ Paste the picture onto a Control Lab project page (⌘ - V).

MS-DOS users: ▶ Load the PCX file into the Contol Lab Shape list by typing `loadpict "filename.pcx` in the Command Center.

A picture box is automatically created and the new shape appears in the box. The new shape is also automatically added to the picture tool Shape list and will appear with a default name, such as `shape1`, on the popup menu.

▶ Add the new shape to any page by using the picture tool and selecting the new shape from the Shape popup menu.

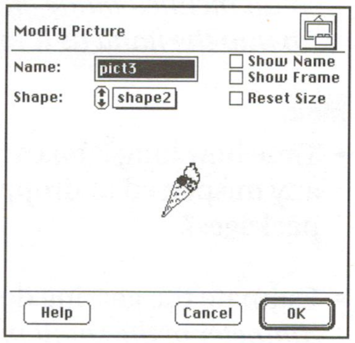


Modify Picture

Name: pict5

Shape: shape1

☐ Show Name
☐ Show Frame
☐ Reset Size



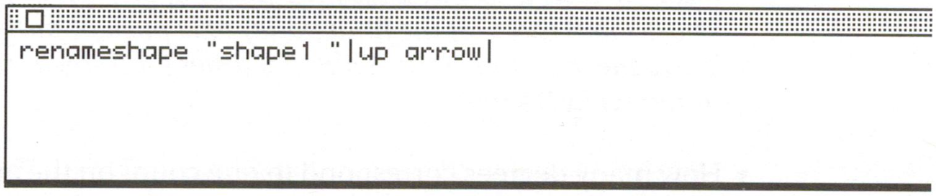
Modify Picture

Name: pict3

Shape: shape2

☐ Show Name
☐ Show Frame
☐ Reset Size

▶ Use the command `renameshape` to change the default names for new shape names to something more meaningful.



```
renameshape "shape1 "|up arrow|
```

Control Lab treats text inside vertical bars || as one word.

The name `up arrow` replaces `shape1` in the picture tool Shape popup menu.

Related Concepts and Extensions (Interdisciplinary)

Technology

- Modify the robot hand to pick up and sort other objects.

- Challenge students to add a motorized conveyor (such as the one shown in card 9701-6) to their robot. An example of the two systems together is shown in a photograph on the back of the 9701 card with the photograph of the Technology set on it.

- Belt drives are included in the arm and hand mechanisms of the robot. What function do the belt drives play? *The belt drive in the turning mechanism is a safety feature. When the robot turns into the chute area by accident, the belt slips, preventing a dangerous situation. The belt drives in the grasping mechanism allow the robot to grasp an object firmly by operating the motor for a short time after the hand has closed on an object.*

Science

- Sometimes a package might fall from the grasp of the robot. Why do you think this happens? *A package falls because there is not enough friction between the fingers of the robot hand and the package to hold it securely.*

- How does the chute area work? *Packages slide down an inclined plane to the stop at the bottom. When the hand grasps a package, it pulls the package to the side at the same time so it clears the stop. A small inclined plane on the edge of the chute helps force the package up into the hand as it tightens.*

Math

- Time how long it takes for the robot to sort 10 objects. Do not count any misplaced or dropped objects. What is the hourly rate for sorting packages?

- Estimate the gearing down ratio of the grasping mechanism. *The diameter of the small pulley wheel is about 1/3 the size of that of the large pulley wheel. This gives an approximate 3:1 gearing down ratio at the motor, not counting slippage. The two gears have the same number of teeth, so they do not affect the overall ratio. The second set of pulley wheels introduces another factor of 3:1, making the total 9:1. The 8-tooth gears on the final pulley axle push or pull the gear racks. Thus, the motor must turn about 9 times for the gear racks to advance or retreat by 8 teeth.*

- How many degrees correspond to one count on the angle sensor? *The angle sensor is actuated by the large 56-tooth outer turntable gear turning an 8-tooth gear on an axle inserted in the sensor. One* *complete turn of the robot would turn the sensor axle seven times. Each turn of the sensor axle registers 16 counts. This means that each count of the sensor axle corresponds to 3.2 degrees (360 / (7 * 16) = 3.2 degrees).*

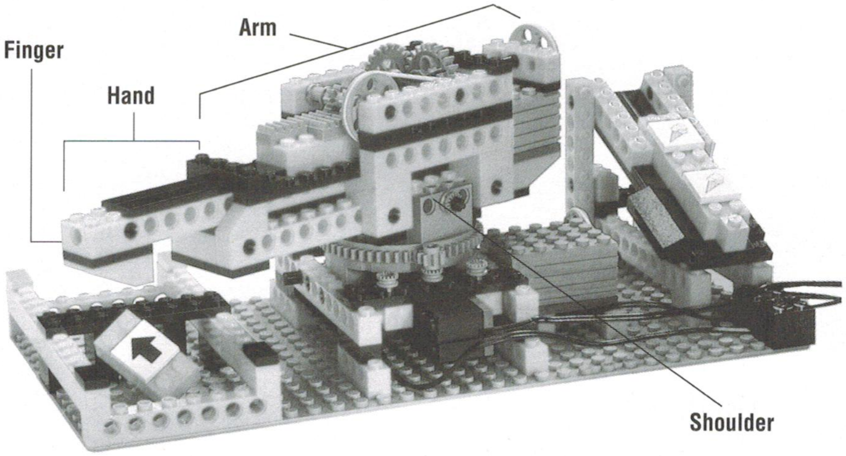
Language Arts

- Draw, scan, and copy a logo for the ice cream company and add it into the control panel (see the optional activity on adding the ice cream and up arrow picture shapes to the control panel after the Programming Ideas section.)

- Write a brief set of robot operating instructions for a new employee.

- Write a poem or short story about a robot.

- Identify the arm, hand and fingers of the LEGO DACTA® Control Lab robot model from building instructions 9701-7.



Social Studies

- Some people argue that robots will take away jobs from people. What types of jobs do robots perform better than people? In what types of jobs do people perform better than robots?

- Think about the type of work you did designing, building and programming the robot. What new jobs might be created by increasing the use of robot technology?

- The growth of the automobile industry allowed profound changes to occur in our economy and social structure. What impacts did the automobile have on the types of jobs created? What impacts did the automobile have on the industries related to horse-powered vehicles? What impacts did the automobile have on where people lived and worked?

Motorized Car and Dynamometer:
A Transportation Technology Investigation

Student Name: ______________________________ **Date:** _______________

Background Information

What is transportation technology?

Transportation technology moves people and products from one point to another. Transportation can be on land (e.g. cars, buses, trains, trucks), in air (e.g. airplanes of many types) in water (e.g. sailing ships, freighters, submarines) and in space (e.g. manned and unmanned shuttles).

Vehicles are important modes of transportation. Many people use a car to travel to and from work or school. Some companies use cars as a means of delivering goods — just think about the number of pizzas delivered every day!

What is speed and how is speed measured?

Speed is the measurement of how far something travels within a certain amount of time.

How speed is measured depends upon the situation. The average speed of a car, for example, can be measured by the distance it travels in miles or kilometers within an hour's time. For example, if it takes three hours for a car to travel 150 miles, the average speed of the car is 50 miles per hour.

A *speedometer* is an instrument that measures the instantaneous speed of a car in miles per hour (MPH) or kilometers per hour (KPH). Some cars have other instruments that can measure the average speed per hour traveled during a trip.

You can use a LEGO DACTA® Control Lab angle sensor to measure the revolutions of a car axle per minute (RPM). An instrument that measures RPM is called a tachometer.

The speed of other things are measured in different units of distance and time. For example, the speed of light is measured in miles or kilometers per second.

How is light measured?

Headlights are crucial for driving at night. The headlights throw a beam of light in front of the car as it moves, illuminating the road for the driver and warning oncoming traffic that another car is on the road.

To work properly, the headlights of the automobile must be positioned correctly. By measuring the intensity of the light at a specific point, you can move the headlight into its optimum position.

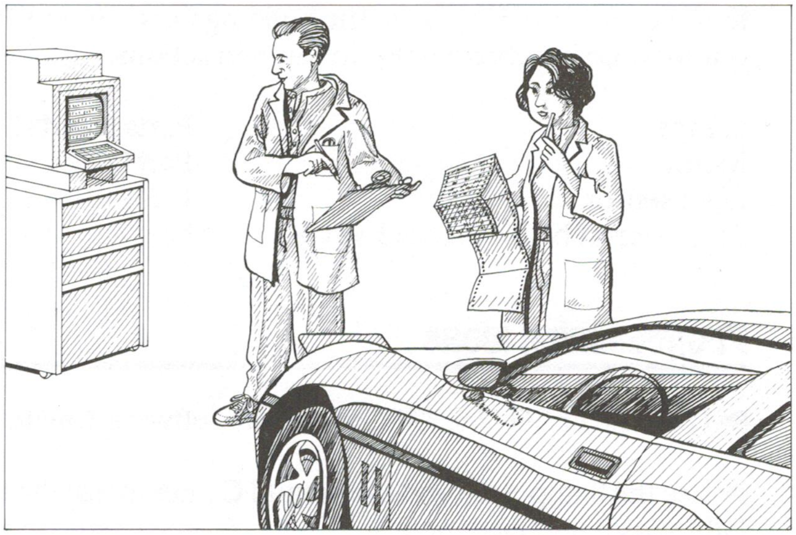
You can use the LEGO DACTA® Control Lab light sensor to measure the light intensity of the lamp.

What is a dynamometer?

To test the performance of new car designs, people who work on cars use dynamometers. The dynamometer holds the car stationary while parts of the car are inspected or a computer analysis of the performance is conducted. The dynamometer allows the car engine to run and the wheels to move so the car speed can be measured.

Speed can be measured using a stopwatch to time a car as someone drives it a specific distance. However, the process is more involved and subject to many potential errors.The dynamometer makes measuring car speed easier and more accurate.

Define The Problem



Problem Statement

Tracy and Jamie design performance automobiles. They need a system to measure the speed of the cars. Also, since the vehicles race at night, the headlights must be checked for proper alignment.

Design Brief

▶ Build a motorized car and a dynamometer to measure the car speed and check the car headlight alignment. Program a control panel that operates the dynamometer.

The dynamometer system must include the following:

- A platform to raise the car from the ground so that the car stays in place as it runs.
- A device for measuring the RPM of the tires.
- A device for checking the alignment of the headlights.
- A computer-controlled button to turn on and turn off the car motor.
- A procedure for calculating the RPM of the tires.
- An area for displaying the RPM of the tires.
- A computer-controlled slider for changing the power level of the motor.
- Computer-controlled buttons to turn on and turn off the headlights.
- An area of the computer control panel for displaying the light intensity of the headlights.
- A computer-controlled slider to adjust the power level of the headlights.

Building Ideas

▶ Build the model shown in the building instructions 9701-3. Attach the outputs and inputs as shown in the instructions.

The optional components are the sound element shown attached to port A and the angle sensor shown attached to port 5 in the building instructions for 9701-3 page 15.

Lamps	Ports B and C
Motor	Port D
Light sensor	Port 6
Angle sensor below the back wheels	Port 7

Programming Ideas

▶ Load the LEGO DACTA® Control Lab software. Create a New Project.

▶ Use the Setup page and Command Center to test the lamps, motor, light sensor and angle sensor.

▶ While testing the system, find out:

- What is the angle sensor count for one minute?
- What are the revolutions per minute?

The angle sensor counts 16 for each revolution.

The circumference of the dynamometer tire on the angle sensor axle and the tire on the car are different. The circumference of the tire affects the speed of the car. In order to calculate the car speed using the dynamometer, you must find out by what to multiply the dynamometer (`angle7`) RPM to find the car RPM.

- Measure the diameter of the two types of tires.
- Calculate the circumference of the two tires.
  $C = \text{diameter} * \pi = \text{diameter} * 3.14$
- Divide the circumference of the smaller dynamometer tire by the larger car tire to find the ratio.
- Multiply the result times the `angle7` feedback to calculate the RPM of the car.

▶ Create a control panel to operate the dynamometer. Review the design brief specifications so that you include important control features.

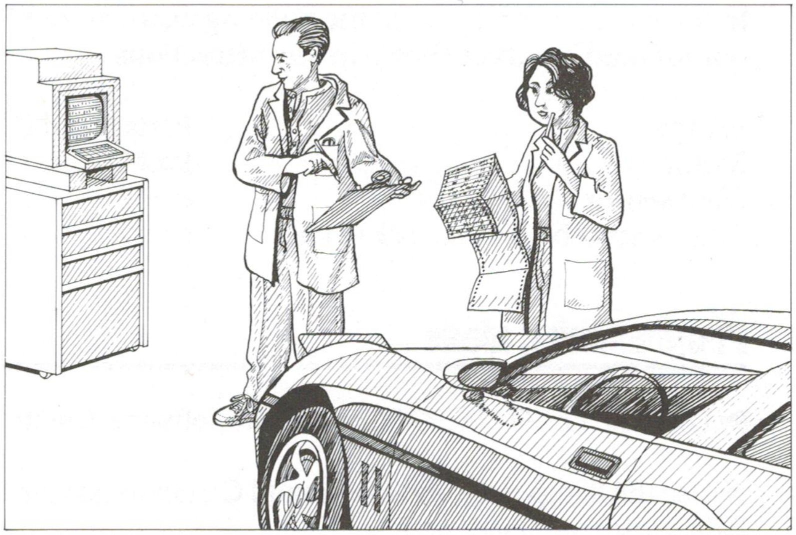
If you do not wish to turn on the dynamometer for one minute to calculate RPM, turn it on for 6 seconds and multiply the angle sensor result by 10.

If the angle sensor reports in negative numbers, flip the motor plug around so the motor moves in the opposite direction. Or, use the primitive `abs` to display the absolute value of any calculation involving `angle7` (e.g. `show abs angle7 * 8`).

Use the primitive `round` to display only the result of any calculation rounded to the nearest integer (e.g. `show round angle7 * 8`).

See the *Reference Guide* for more information about and examples of `abs` and `round`.

Define The Problem



Problem Statement

Tracy and Jamie design performance automobiles. They need a system to measure the speed of the cars. Also, since the vehicles race at night, the headlights must be checked for proper alignment.

Design Brief

▶ Build a motorized car and a dynamometer to measure the car speed and check the car headlight alignment. Program a control panel that operates the dynamometer.

The dynamometer system must include the following:

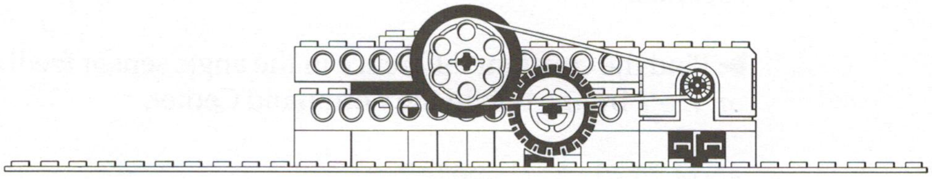
- A platform to raise the car from the ground so that the car stays in place as it runs.
- A device for measuring the RPM of the tires.
- A device for checking the alignment of the headlights.
- A computer-controlled button to turn on and turn off the car motor.
- A procedure for calculating the RPM of the tires.
- An area for displaying the RPM of the tires.
- A computer-controlled slider for changing the power level of the motor.
- Computer-controlled buttons to turn on and turn off the headlights.
- An area of the computer control panel for displaying the light intensity of the headlights.
- A computer-controlled slider to adjust the power level of the headlights.

Applied Math and Science Investigation: Comparing the Motorized Car and Dynamometer RPM

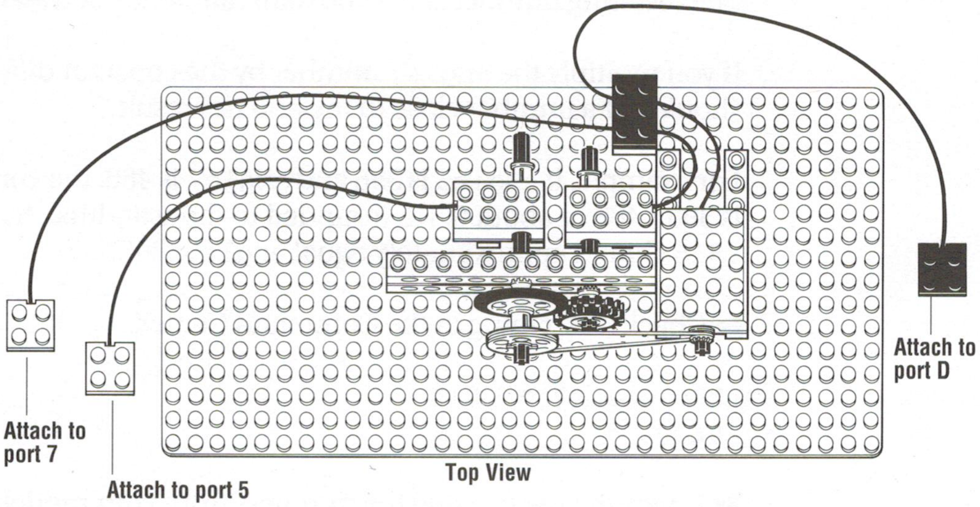
The mechanism shown below can be used to calculate the actual RPM difference in the larger car tire and smaller dynamometer tire.

Building Ideas

▶ Build the mechanism as shown.



Front View



Top View

▶ Attach the angle sensor connected to the larger tire axle to port 5.

▶ Attach the angle sensor connected to the smaller tire axle to port 7.

▶ Attach the motor to port D.

The motor is mounted on the opposite side of the larger tire pulley so there is enough friction between the tires to make them move.

In the mechanism shown above, the motor turns the larger tire, just as the car motor turns the larger tire on the car. The larger tire on the mechanism is designed to force the smaller tire to rotate, just as the car tire forces the dynamometer tire to rotate. The angle sensors report the rotations of both tire axles.

Programming Ideas

Double-click on each icon to open the dialog box. Then click Reset Sensor. Click OK to close the dialog box. Or, type `resetrotation 5` `resetrotation 7` in the Command Center.

▶ Start LEGO DACTA® Control Lab software. Create a New Project.

▶ Drag a motor icon to port D. Drag an angle sensor icon to port 5 and port 7.

▶ Make sure `angle5` and `angle7` are reset to 0.

▶ Reduce the power level of the motor by typing `talkto "motord setpower 6` in the Command Center.

▶ Click in one of the circles above the `motord` icon and hold for a few seconds.

▶ Find the constant difference in the angle sensor feedback by dividing `angle5` by `angle7` in the Command Center.

```
show angle5 / angle7
```

The resulting number is the constant difference between the two tires.

If you multiply the `angle7` number by the constant difference, you should get a number close to the `angle5` result.

For example, if `angle5` is 396 and `angle7` is 486, the constant difference is 396 / 486 or about 0.815 (rounded to thousandths). So, `angle7 * .815` should approximately equal `angle5` (396).

▶ Check it by typing in the Command Center.

```
show angle7 * .815
396.09
```

▶ Check the result using the numbers from your model.

You have calculated the actual difference between the car tire and dynamometer tire which includes a factor for friction and belt slippage.

Teacher Notes for Motorized Car and Dynamometer: A Transportation Technology Investigation

✏ Teacher note:

See the "What Do Students Learn?" section in the beginning of this book for concepts in technology, science, math, language arts and social studies that are addressed by the Dynamometer activity.

Objectives

The purpose of this investigation is to introduce students to transportation technology systems through building and programming a motorized car and dynamometer. By the end of the project, students will be able to:

1. Describe the process they used to develop the dynamometer.
2. Explain how distance and speed can be measured.
3. Explain how light can be measured.

Students will also develop specific skills working with the LEGO DACTA® Control Lab output and input devices and control features. By the end of the project, students will be able to:

1. Control the movement of one power source (motor).
2. Use pulley and gear ratios to reduce the speed of the power source.
3. Use software tools and commands to develop a control panel for the dynamometer.
4. Use feedback from sensors to monitor the car speed and headlight alignment of a vehicle using the Control Lab software and the control panel.

Materials Needed

- One LEGO DACTA Technology Set and building instruction 9701-3 for each group of 2 - 3 students.
- At least one Control Lab system.
- Photocopies of the student Motorized Car and Dynamometer pages.
- Disks or disk storage space for each student group project.
- Copies of the Measurement Worksheet from the Supplemental Materials section of this book (optional).
- Copies of the Applied Math and Science Investigation page. The copymaster is located after the student Building and Programming Ideas page (optional).

Background Information

✏ **Teacher note:**

Students should probably try the Applied Math and Science Investigation (student pages 3.5-3.6), after reading the Background Information and before attempting to develop a solution for the design brief.

To prepare students for the dynamometer activity, two pages of background information provide answers to the following questions: What is transportation technology? What is speed and how is speed measured? How is light measured? What is a dynamometer?

See the Bibliography in the Supplemental Materials section of this book for more information on transportation technology.

Since students must use light sensor and angle sensor feedback in the dynamometer system, they may need to review "Exploration 4" from the *Setup Guide and Introductory Explorations* and *Quick Start Guide Appendix A*, "Using the Angle Sensor" prior to programming the dynamometer.

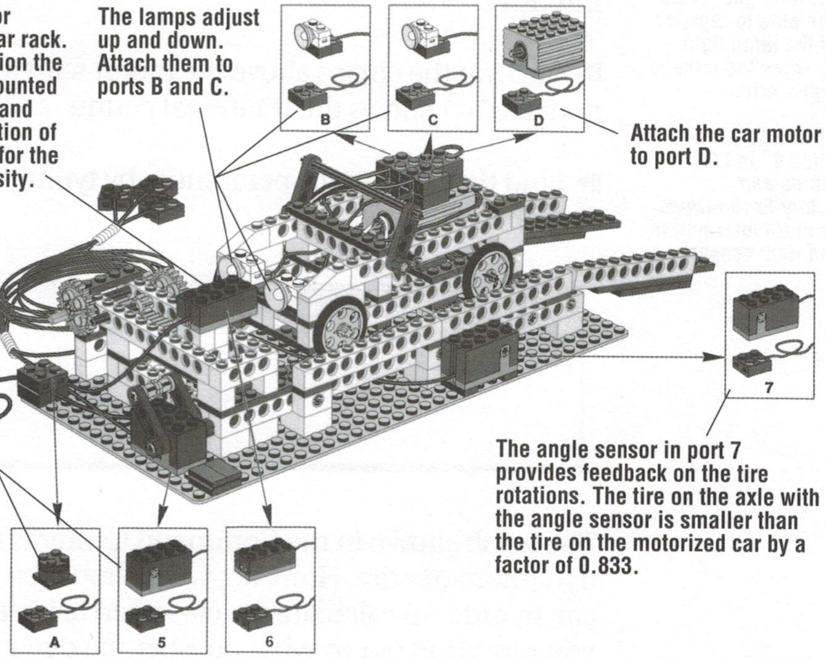
There are many math applications within this activity. You may wish to extend the available class time by teaching with a math teacher or other interested educator.

Define The Problem

Several criteria are listed on the student design brief page. You may wish to add your own.

Building Ideas

Students can follow the step by step building instructions labelled 9701-3 to create this dynamometer model or they can design and build their own.



The light sensor moves on a gear rack. Manually position the light sensor mounted on a gear rack and adjust the position of the headlights for the optimum intensity.

The lamps adjust up and down. Attach them to ports B and C.

Attach the car motor to port D.

The angle sensor in port 7 provides feedback on the tire rotations. The tire on the axle with the angle sensor is smaller than the tire on the motorized car by a factor of 0.833.

The sound element in port A and the angle sensor in port 5 are optional. The solution shown in these teacher notes do not use them.

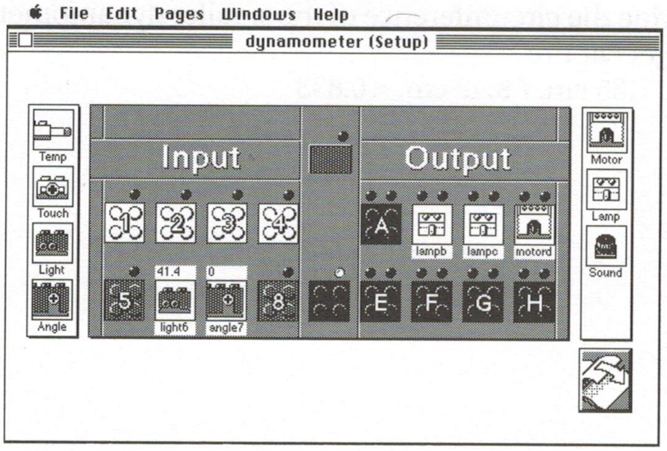
Programming Ideas

▶ Load the LEGO DACTA® Control Lab software. Create a New Project.

▶ Make sure the lamps, motor, light sensor and angle sensor are attached to the LEGO DACTA® Serial Interface as shown in the building ideas.

▶ Drag the lamp icons to ports B and C. Drag the motor icon to port D.

▶ Drag the light sensor icon to port 6 and the angle sensor icon to port 7.



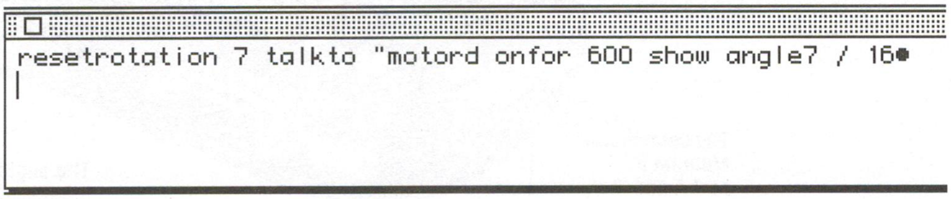
Light intensity is measured as a percentage. At this distance, the light sensor should be able to register 100% of the lamp light intensity when the lamp is aligned properly.

See "Introductory Exploration 4" in the *Setup Guide and Introductory Explorations* book for more information about the light sensor.

▶ Click on the circles above the lamp icons one at a time and move the light sensor in front of each of them to make sure it is registering the light intensity.

▶ Click on the circles above the motor icon to make sure the car moves and `angle7` shows the rotational counts.

▶ Find the revolutions per minute by typing the following in the Command Center.



```
resetrotation 7 talkto "motord onfor 600 show angle7 / 16•
```

The result shown in the Command Center is the RPM of the dynamometer tire. However, what should be measured is the RPM of the car. In order to calculate the car speed using the dynamometer feedback, you must find out by what number the dynamometer feedback (angle7) must be multiplied in order to calculate car RPM.

This is the theoretical difference between the tires because friction and other factors are not taken into account.

Students can find the actual difference between the two tire sizes by doing the activity "Applied Math and Science Investigation: Comparing the Motorized Car and Dynamometer RPM." A copymaster can be found after the student Building and Programming Ideas page.

Students could compare diameter rather than circumference to find the ratio; however, using circumference can lead to discussions about measuring speed in centimeters per second. See the Science and Math extensions: the fourth one shows how to measure speed in centimeters per second using the `find.cm.sec` procedure.

▶ Find the difference between the two tires by comparing the circumference.

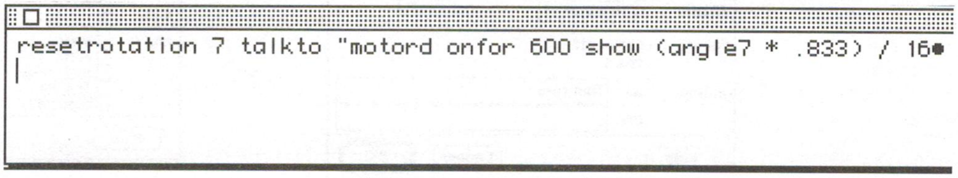
- Measure the diameter of the two tires.
  Dynamometer tire diameter = 2.5 cm.
  Car tire diameter = 3.0 cm.

- Calculate the circumference of both tires.
  C = Diameter * π = Diameter * 3.14
  Dynamometer tire circumference = 2.5 cm. * 3.14 = 7.85 cm.
  Car tire circumference = 3.0 cm. * 3.14 = 9.42 cm.

- Divide the circumference of the smaller dynamometer tire by the larger car tire.
  7.85 cm. / 9.42 cm. = 0.833

- Multiply the `angle7` feedback by the result to calculate the RPM of the car.

The parentheses shown here are optional.



```
resetrotation 7 talkto "motord onfor 600 show (angle7 * .833) / 16●
```

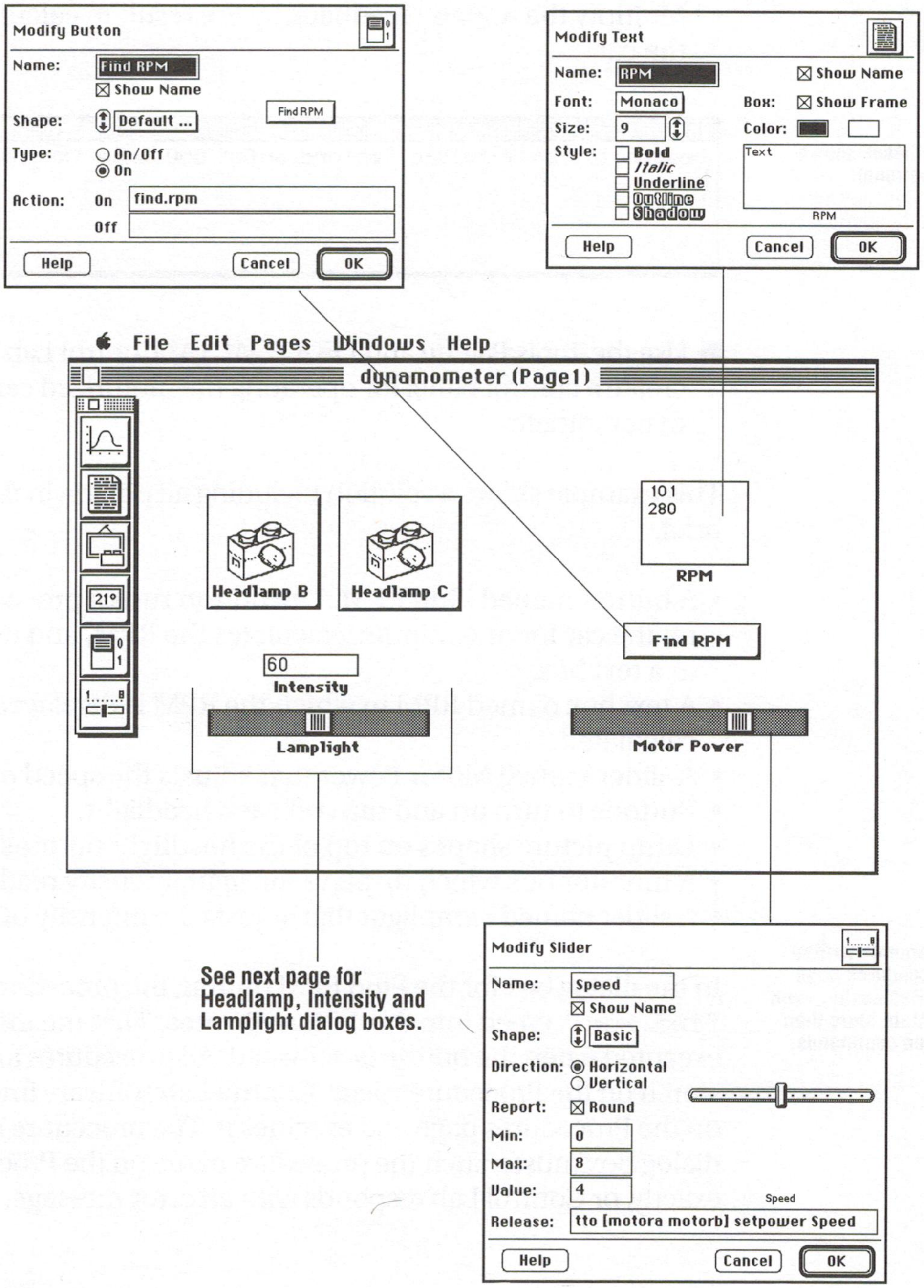
▶ Use the Tools Palette and LEGO DACTA® Control Lab commands to create a control panel for operating the motorized car and dynamometer.

This example shows a solution including all criteria in the student design brief.

- A button named Find RPM. The button runs a procedure which turns on the car for one minute, calculates the RPM, and displays the result in a text box.
- A text box named RPM in which the RPM is displayed after it is calculated.
- A slider named Motor Power that adjusts the speed of the car motor.
- Buttons to turn on and turn off each headlight.
- Lamp picture shapes on top of the headlight buttons for illustration.
- A monitor box which displays the light intensity readings.
- A slider named Lamplight that adjusts the intensity of the headlights.

Logo statements written on the Procedures page can be edited easily, even if they contain more than two or three commands.

In the dialog box for the Find RPM button, the procedure name `find.rpm` is typed into the Action: On area. That means the procedure is executed when the button is activated. All procedures are typed and stored on the Procedures page. Control Lab software finds the procedure on the Procedures page and executes it. The procedure name used in the dialog box must match the procedure name on the Procedures page exactly, or Control Lab responds with an error message.



Modify Button

Name: Find RPM
☒ Show Name
Shape: Default ...
Type: ○ On/Off ● On
Action: On find.rpm
Off

Modify Text

Name: RPM ☒ Show Name
Font: Monaco Box: ☒ Show Frame
Size: 9
Style: ☐ Bold ☐ Italic ☐ Underline ☐ Outline ☐ Shadow

Modify Slider

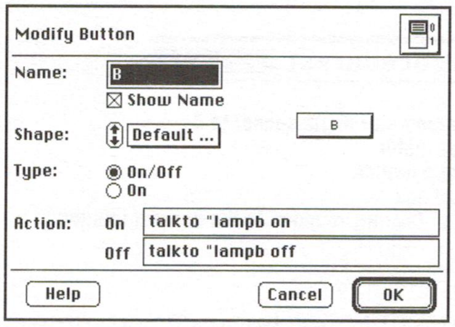
Name: Speed
☒ Show Name
Shape: Basic
Direction: ● Horizontal ○ Vertical
Report: ☒ Round
Min: 0
Max: 8
Value: 4
Release: `tto [motora motorb] setpower Speed`

See next page for Headlamp, Intensity and Lamplight dialog boxes.

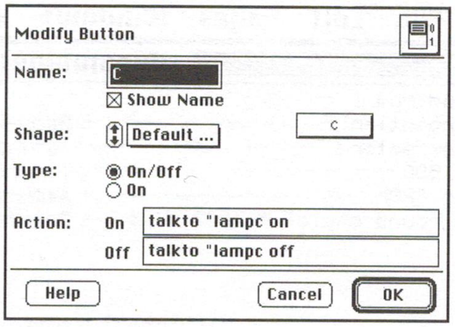
LEGO DACTA® Control Lab treats two or more words enclosed in vertical bars | | as one word.

The minimum value set for the Motor Power slider is 4 because at power levels from 1 to 3 the car moves very slowly or not at all. `Setpower` will take an input from 0 to 8.

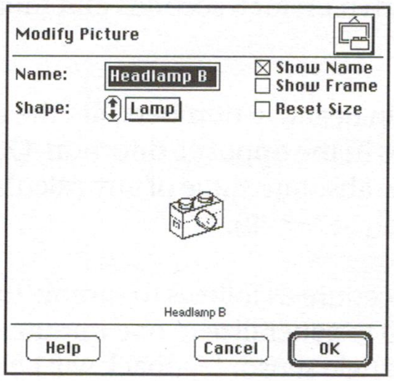
The buttons that turn on and turn off the lamps are named B and C. These short names allow the lamp pictures placed on top of the buttons to cover the names. Deselect the Show Name option in the B and C dialog boxes to create "hidden buttons" — buttons with no name or frame showing — just the lamp pictures over them.


Modify Button
Name: B
☒ Show Name
Shape: Default ...
Type: ● On/Off ○ On
Action: On talkto "lampb on
Off talkto "lampb off

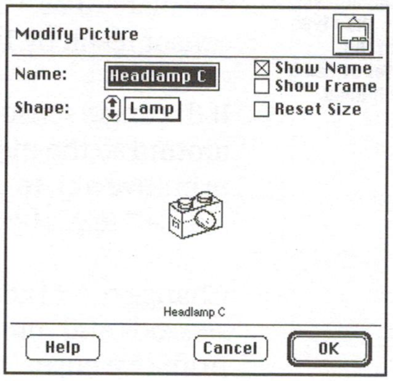
Lamp B button dialog box


Modify Button
Name: C
☒ Show Name
Shape: Default ...
Type: ● On/Off ○ On
Action: On talkto "lampc on
Off talkto "lampc off

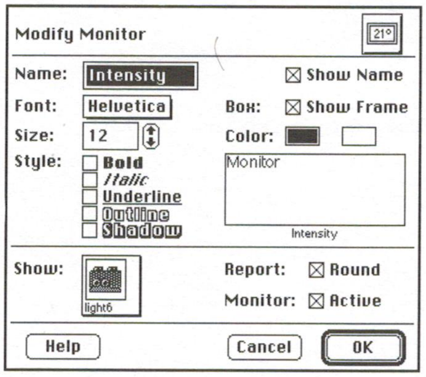
Lamp C button dialog box


Modify Picture
Name: Headlamp B
Shape: Lamp
☒ Show Name
☐ Show Frame
☐ Reset Size

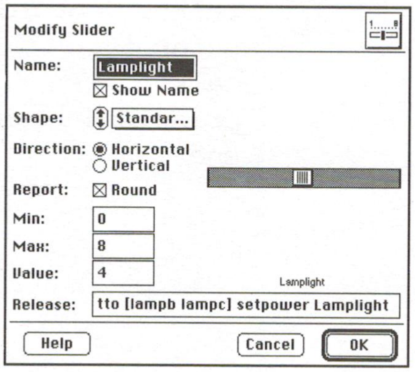
Headlamp B picture shape dialog box


Modify Picture
Name: Headlamp C
Shape: Lamp
☒ Show Name
☐ Show Frame
☐ Reset Size

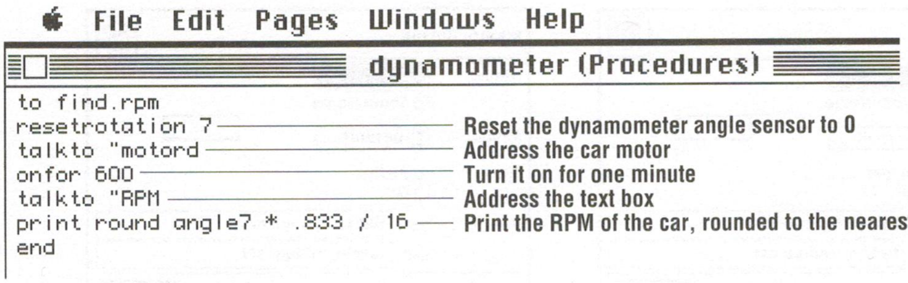
Headlamp C picture shape dialog box


Modify Monitor
Name: Intensity
☒ Show Name
Font: Helvetica
Box: ☒ Show Frame
Size: 12
Color:
Style: ☐ Bold ☐ *Italic* ☐ Underline ☐ Outline ☐ Shadow
Show: light6
Report: ☒ Round
Monitor: ☒ Active

Intensity monitor dialog box


Modify Slider
Name: Lamplight
☒ Show Name
Shape: Standar...
Direction: ● Horizontal ○ Vertical
Report: ☒ Round
Min: 0
Max: 8
Value: 4
Release: tto [lampb lampc] setpower Lamplight

Lamplight slider dialog box



```
to find.rpm
resetrotation 7
talkto "motord
onfor 600
talkto "RPM
print round angle7 * .833 / 16
end
```

Reset the dynamometer angle sensor to 0

Address the car motor

Turn it on for one minute

Address the text box

Print the RPM of the car, rounded to the nearest integer

Important: `Cleartext` **erases text including the Procedures page. Students must make sure they are addressing ("talking to") the correct text area before they type** `cleartext` **in the Command Center or run a procedure which uses** `cleartext`**.**

To clear the text from the RPM text box on Page1, type `talkto "RPM cleartext` in the Command Center.

If students do not wish to turn on the dynamometer for one minute to calculate RPM, they can turn it on for 6 seconds and multiply the angle sensor result by 10.

If the angle sensor reports in negative numbers, flip the motor plug around so the motor moves in the opposite direction. Or, use the primitive `abs` to display the absolute value of any calculation involving `angle7` (e.g. `show abs angle7 * 8`).

Change the `find.rpm` procedure as follows to sample for only six seconds and multiply by 10 to generalize to one minute. The `abs` primitive ensures that the angle sensor feedback will be positive rather than negative.

```
to find.rpm2
resetrotation 7
talkto "motord
onfor 60
talkto "RPM
print abs round ((angle7 * .833) / 16) * 10
end
```

Related Concepts and Extensions (Interdisciplinary)

Technology

- Add a button to the control panel for switching between “bright” and “normal” headlight lamp intensity.

- Use illustration board behind the light sensor to mark reference lines where the headlight beams focus when they are aligned properly.

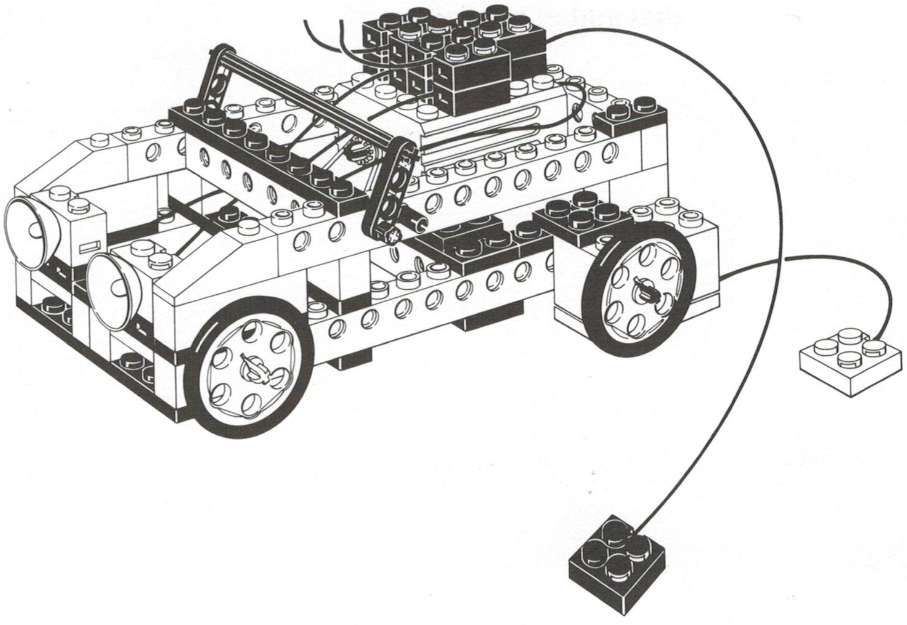
- Program the angle sensor attached to port 5 (see the building instructions 9701-3 page 15) to act as a throttle for the car motor.

Science and Math

- Add mass to the vehicle and measure the effect on the car speed. Graph the speed at different power levels over one minute and compare the effect of the different car masses.

- Use the car motor as a generator to turn on the headlights. Detach the car from the LEGO DACTA® Serial Interface. Attach the connector plates of the lamp wires to the connector plate of the motor wire. Roll the car along the desk or floor. The headlights turn on. Why? *The kinetic energy of the car turns the wire coil inside the motor. Since the wire coil is inside a magnetic field, electricity flows through the motor and the motor wire and turns on the lamps. The motor becomes a generator.*

- Modify the motorized vehicle to include an angle sensor on its back axle. Program the car to move a specified distance, then calculate the speed. You can attach several connecting leads together so the car has enough wire to move along the floor or a table.



✏ Teacher note:

If students use the `find.cm.sec` prodedure with the angle sensor mounted on the motorized car (see previous page for illustration), they can eliminate the `* .833` part of the equation.

- Modify the `find.rpm` procedure to calculate the average speed in centimeters per second. The solution shown below multiplies the total number of revolutions `((angle7 * .833) / 16)` by the distance in centimeters per revolution — the circumference of the tire — 9.42 centimeters. That result is the total distance moved in centimeters. Then the total distance in centimeters is divided by six seconds (60 tenths of a second) to find the average distance moved in centimeters per second.

```
to find.cm.sec
resetrotation 7
talkto "motord
onfor 600
getpage "page1
talkto "|Speed CM Per Sec|
print round (((angle7 * .833) / 16) * 9.42) / 60
end
```

- See the additional student page, "Applied Math and Science Investigation: Comparing the Motorized Car and Dynamometer RPM" for another math and science extension. In this extension, a model to measure the actual difference between the car and dynamometer tire RPM is built and programmed.

Language Arts and Social Studies

- Write a one minute radio or television advertisement for the car. Before preparing the advertisement, describe the demographics (e.g. age range, income level) of the group to which the car should appeal. In the advertisement itself, choose key selling features that appeal to that audience.

Color Code Reader: A Communication Invention

Student Name: ______________________________ **Date:** ______________

Background Information

What is Communication Technology?

Communication technology is the process of transmitting and receiving information. The two main types of communication are graphic and electronic. Using LEGO DACTA® Control Lab involves both graphic and electronic communication.

What are graphic and electronic communication?

In graphic communication, printed words or images are used. Examples of graphic communication include writing, printing, photocopying and drawing.

In electronic communication, electrical signals are transmitted. Examples of electronic communication include using the telephone, television, satellite and radio.

Features of Control Lab such as the Setup page, buttons, sliders, graphs and text boxes are forms of graphic communication. The information exchanges among the computer mouse, keyboard, monitor, LEGO DACTA® Serial Interface and sensors involve electronic communication.

What is a code?

Computers and other electronic communication devices send messages back and forth using codes. The computer is based on a digital code — a series of 0's and 1's — which is programmed to be interpreted or "decoded" a specific way. The way the information is coded and decoded depends upon the type of computer hardware and software used.

The Control Lab system uses the Logo programming code. Both you and the computer use to communicate with each other. For example, the commands `talkto "motora on` are Control Lab Logo commands that mean "address a motor attached to output port a and turn it on."

How are codes used?

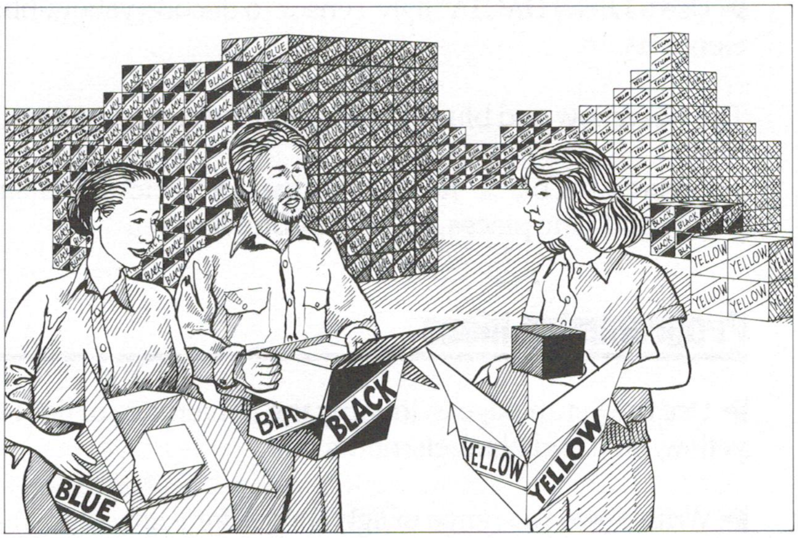
A computer system is flexible because the type of system used determines how the information is coded and decoded.

You begin to code information by using the keyboard, mouse and sensors to input into the computer. The Control Lab software and computer hardware accept the input and further code the information by turning it into 0's and 1's (digital code). Then the system transmits the information along the computer circuitry. The system then decodes the information and outputs a response. The response can be a reaction by a LEGO DACTA® Control Lab output device (e.g. a motor turns on) or by a Control Lab screen object (e.g. a text box displays a message).

By creating a control panel using the Control Lab software, you can design a communication system that encodes, transmits, receives and decodes input from the computer keyboard, computer mouse or Control Lab sensors. The computer system can also store and retrieve the information for later use or print the information on paper.

In this activity, you will use a sensor to code and decode colors (electronic communication) and a control panel to display the sensor input and the color detected (graphics communication).

Define the Problem



Problem Statement

Arla and Jack are disappointed in the new high speed robotic spray painting system they have installed in their factory. After receiving complaints from customers, they discovered that several items are not being spray painted correctly. They need a system that checks each item after it passes the spray painting booth and identifies the items that are not painted properly.

Design Brief

▶ Design an optical system that checks the color of an item. Program the control panel to operate the system.

The color code reading system must include the following:
- A sensor that checks the color of an item.
- The ability to distinguish among three colors.
- Controls for checking and reporting the color detected.
- A signal when the "incorrectly sprayed" item is detected.

Additional Criteria

The code reader design can also include:
- An auditory signal to identify correctly painted items.
- A different auditory signal to identify incorrectly painted items.

Building Ideas

▶ Use a LEGO DACTA® light sensor to decode yellow, blue and black elements.

The tall yellow and blue elements are the easiest to handle.

▶ Create a tall black "item" using five 1 x 2 bricks, similar to the tall yellow and blue pieces in the 9701 building set.

Programming Ideas

▶ Use the Setup page to identify the percentage of light reflected by the yellow, blue and black elements.

The Light Sensor Data Sheet can help you organize your data and Logo procedures.

▶ Write down the range of light values for each color.

▶ Based on your data, decide:
- Which color(s) and corresponding light sensor values indicate correctly spray painted item(s)?
- Which color(s) and corresponding light sensor values indicate incorrectly spray painted item(s)?

▶ Figure out the maximum and minimum values for each color.

▶ Then write a procedure that checks for each color and reports the color detected.

At least one color must be identified by the procedure as an item that is incorrectly spray painted.

▶ On the project page, include the following:
- A button that runs a color-checking procedure.
- A monitor that reports the current light sensor value.
- A text box that displays the color detected by the light sensor and a warning message when an incorrectly sprayed item is detected.

▶ To incorporate the ideas in the additional criteria of the design brief, add a sound element to the model and the following features to the control panel.
- One type of sound when the item is correctly spray painted.
- Another type of sound when the item is incorrectly spray painted.

Light Sensor Data Sheet

Student Name ______________________________

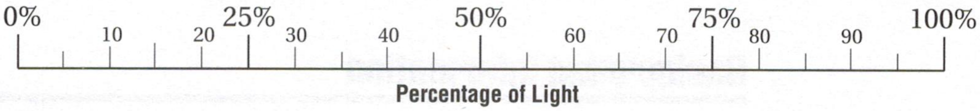
Directions

The distance between the light sensor and the colored item affects the percentage of light data.

- Make sure the light sensor is attached to port 6.
- Place a yellow item in front of the light sensor so that one stud is between them.
- Write down the percentage of light feedback from the light sensor in the first row and column on the chart.
- Continue collecting light sensor data for the yellow item at different distances from the light sensor. Record the data on the chart.
- Then collect data for the blue and black items.
- After collecting the light sensor data for all three colors at different distances, choose the distance for which you will design your control system.
- Figure out the range of light sensor values at that distance for which the computer can be programmed to identify a color as yellow, blue and black. Mark the range on the Percentage of Light Scale below.
- Write the Logo `if` statements that can be used decode the color from the light sensor feedback. See several formats for the `if` statements below as ideas.

It is a good idea to check the same color at the same distance two or three times. Use the Trial 2 and Trial 3 spaces for this.

Color	Distance from sensor	1 Stud	2 Studs	3 Studs	Remarks
Yellow	Trial 1				
	Trial 2				
	Trial 3				
Blue	Trial 1				
	Trial 2				
	Trial 3				
Black	Trial 1				
	Trial 2				
	Trial 3				



Range for yellow__________: Not less than _____ Not greater than _____.

Range for blue___________: Not less than _____ Not greater than _____.

Range for black__________: Not less than _____ Not greater than _____.

See the *Reference Guide* for more information about the `if`, `and`, `>`, `<`, and `=` commands.

```
if light6 > ____ [your procedure here]
if light6 < ____ [your procedure here]
if light6 >= ____ [your procedure here]
if light6 <= ____ [your procedure here]
if and light6 >= ____ light6 < ____ [your procedure here]
```

Teacher Notes for Color Code Reader: A Communication Invention

Objective

The purpose of the Color Code Reader invention is to introduce students to communication systems through designing, building and programming a color code reader. By the end of the project, students will be able to:

1. Describe the process they used to develop the color code reader.
2. Explain why the color code reader is an example of communication technology.

Students will also develop specific skills working with the LEGO DACTA® Control Lab output and input devices and control features. By the end of the project, students will be able to:

1. Design and build a color code reading system.
2. Use software tools and commands to develop a control panel for the color code reader.
3. Control the color code reader using the Control Lab software and a control panel.

✏ Teacher note:
See the "What Do Students Learn?" section at the beginning of this book for concepts in technology, science, math, language arts and social studies that are addressed by the Color Code Reader activity.

Materials Needed

- One LEGO DACTA Technology Set and building instructions for each group (2 - 3 students).
- At least one Control Lab system.
- Photocopies of the student Color Code Reader pages.
- Photocopies of the Light Sensor Data sheet (two copies per group).
- Disks or disk storage space for each student group project.
- For the Extensions (optional): mirror, protractor.

Background Information

To prepare students for the color code reader activity, two pages of background information provide answers to the following questions: What is Communication technology? What are graphic and electronic communication? What is a code? How are codes used?

The Color Code Reader Invention focuses on programming and requires very little building. The Bridge Invention requires more building and, depending upon the type of solution, can require a little or a lot of programming.

See the Bibliography in the Supplemental Materials section of this book for additional references.

✏ Teacher note:
This activity assumes students have tried the *Introductory Explorations 1-4* in the *Setup Guide and Introductory Explorations* book which show how to use the sound element and light sensor (among other output and input devices).

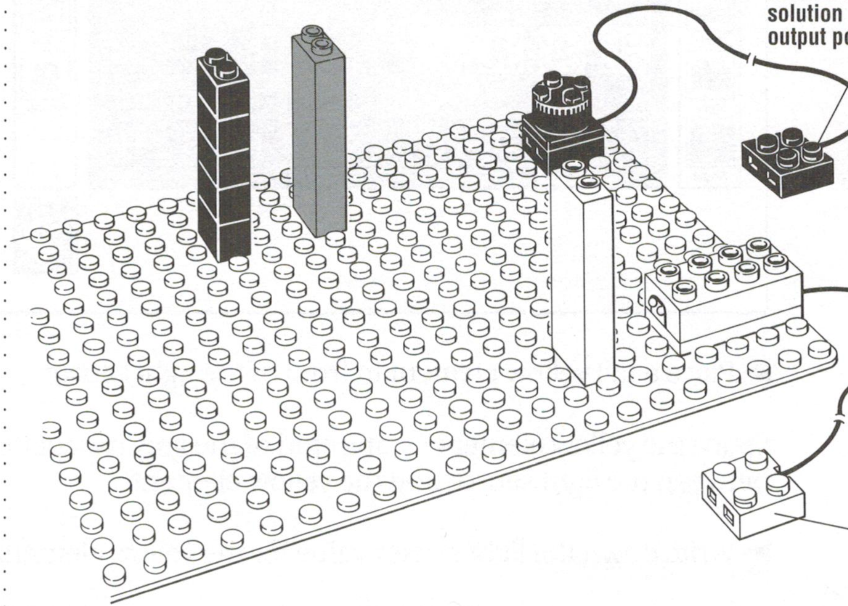
A Control Lab systems overview showing input, process and output can be found on page 5 of the *Setup Guide*.

Define the Problem

Several criteria are listed on the student design brief page. You may wish to add your own.

Building Ideas

There are enough elements in the LEGO DACTA® Technology building set for four tall yellow elements, two tall blue elements and two black items (five 1 x 2 bricks per item).


The sound element is used in the Example B solution only. Attach to output port B.

Attach to input port 6.

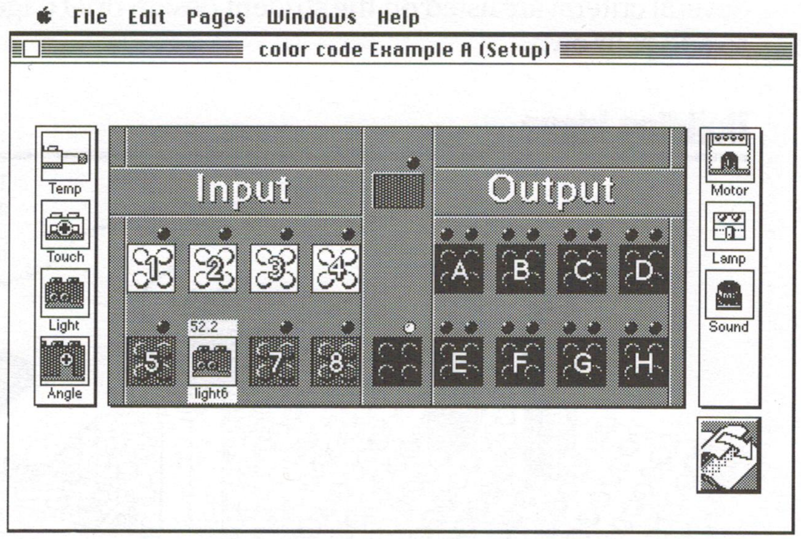
Programming Ideas Example A

This example addresses the main criteria in the student design brief. It shows how to use a button to run a color checking procedure, distinguish among three colors, report the color detected and print a message when an incorrectly sprayed item is detected.

▶ Load LEGO DACTA Control Lab software. Start a New Project.

▶ Attach a light sensor to input port 6 on the LEGO DACTA Serial Interface.

▶ Drag a light sensor icon to input port 6 on the Setup page.



▶ Place a tall yellow element in front of the light sensor.

Space the yellow element so one stud of the grey baseplate shows in between the light sensor and the yellow element.

✏ Teacher note:

A Light Sensor Data Sheet photocopy master is provided on page 4.5 of the student copymasters for the Color Code Reader.

▶ Write down the light sensor value for the yellow element.

▶ Repeat the process for the blue and black elements.

▶ Now, based on the data, figure out how the range of light feedback should be decoded by the computer.

▶ Write Logo `if` statements that the computer will use to identify yellow, blue and black elements.

The code `>=` means greater than or equal to; `<=` means less than or equal to. For more information about these commands and the `and` command, see the *Reference Guide*.

Example A uses the following statements to decode the colors:

Yellow:	`if light6 >= 50`
Blue:	`if and light6 >= 40 light6 <=50`
Black:	`if light6 <= 40`

▶ Decide which items will be considered correctly spray painted and which items will be considered incorrectly spray painted.

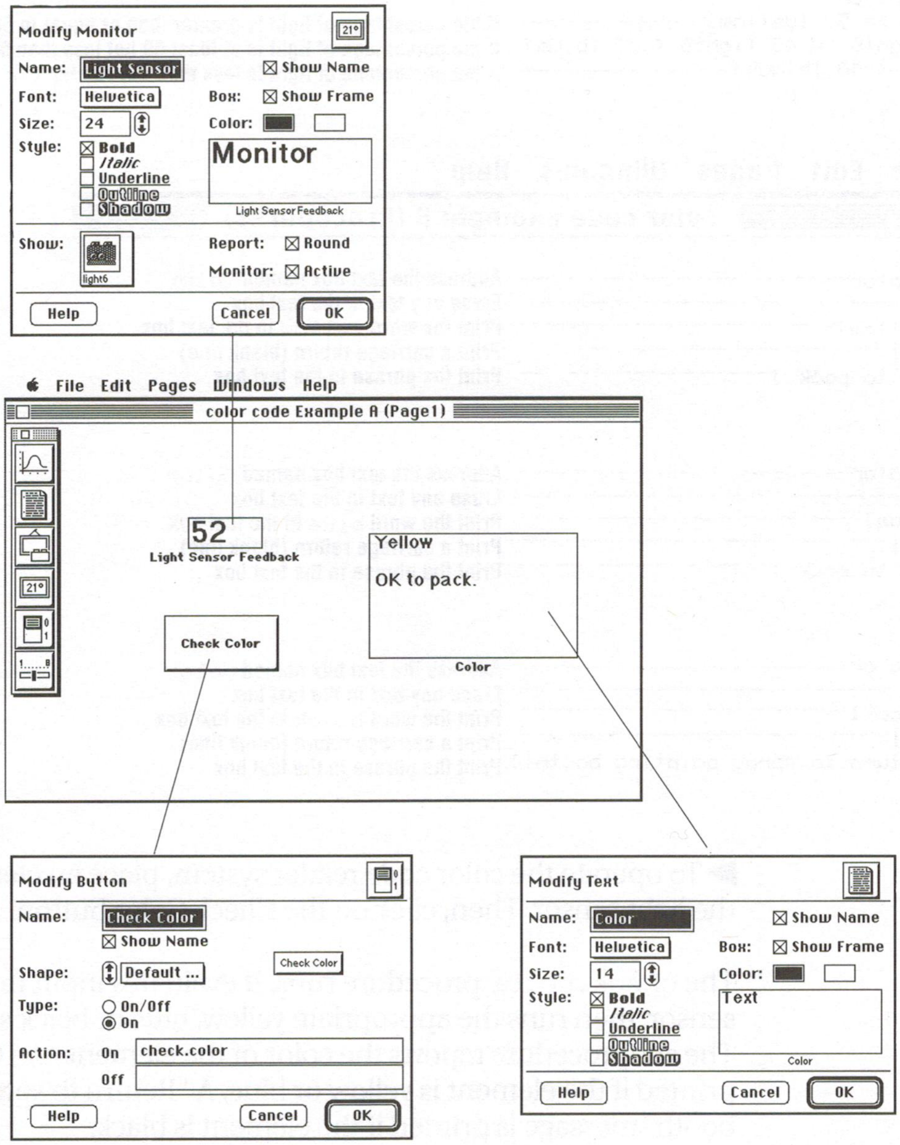
Example A interprets the yellow and blue elements as correctly spray painted items and black elements as incorrectly spray painted items.

▶ Use the Tools Palette and LEGO DACTA® Control Lab commands to create a control panel for operating the color code reader.

Example A shows a control panel with the following features:

- A button which runs a color checking procedure on the Procedures page.
- A monitor box to display the current light sensor value.
- A text box to print the color decoded by the `check.color` procedure.

The text box also displays an OK message if the item is correctly spray painted (yellow or blue elements) and a "Return to spray painting booth" message if the item is not correctly spray painted (black elements).



Modify Monitor

Name: Light Sensor ☒ Show Name

Font: Helvetica Box: ☒ Show Frame

Size: 24 Color:

Style: ☒ Bold ☐ Italic ☐ Underline ☐ Outline ☐ Shadow

Monitor

Light Sensor Feedback

Show: light6

Report: ☒ Round

Monitor: ☒ Active

color code Example A (Page1)

52

Light Sensor Feedback

Yellow

OK to pack.

Color

Check Color

Modify Button

Name: Check Color

☒ Show Name

Shape: Default ...

Type: ○ On/Off ● On

Action: On check.color

Off

Modify Text

Name: Color ☒ Show Name

Font: Helvetica Box: ☒ Show Frame

Size: 14 Color:

Style: ☒ Bold ☐ Italic ☐ Underline ☐ Outline ☐ Shadow

Text

Color

Notice in the dialog box for the Check Color button, the procedure name is typed in the Action: On area. That means the procedure is executed when the button is activated. All procedures are typed and stored on the Procedures page. Control Lab software finds the procedure on the Procedures page and executes it. The procedure name used in a dialog box must match the procedure name on the Procedures page exactly, or Control Lab respond with an error message.

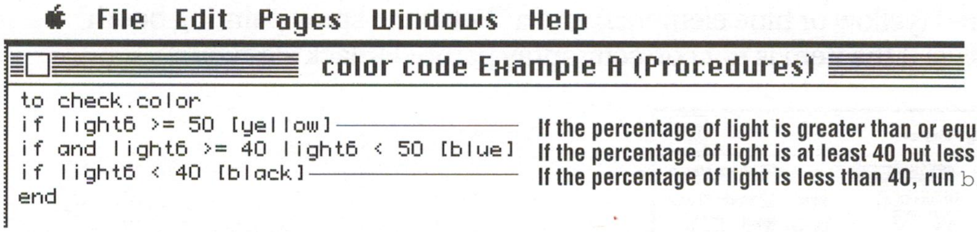
Writing Logo statements on the Procedures page makes writing and editing them much easier, especially if they are more than two or three commands long.

▶ Make a button to run a procedure on the Procedures page that checks the color of an element.

See the *Quick Start Guide* sections "Using the Button Tool," "Changing the Type of Button," "Using the Text Tool" and "Using the Monitor Tool."

▶ Use the monitor tool to make a monitor that displays the light sensor value.

▶ Use the text tool to make a text box for the control panel. Write procedures that print the color detected by the light sensor and any messages in the text box.

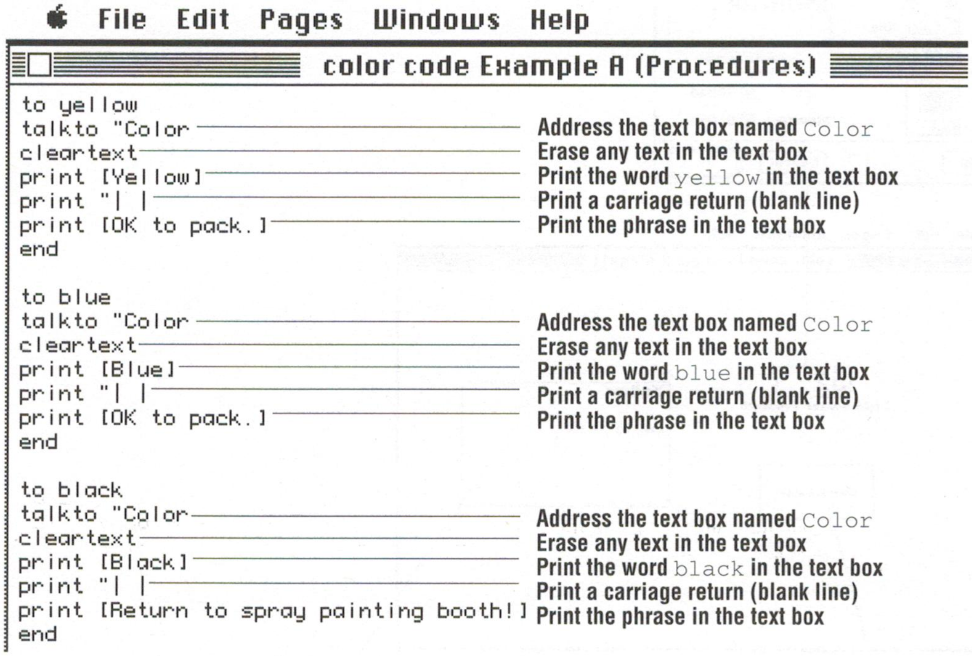


```
to check.color
if light6 >= 50 [yellow]
if and light6 >= 40 light6 < 50 [blue]
if light6 < 40 [black]
end
```

If the percentage of light is greater than or equal to 50, run `yellow`

If the percentage of light is at least 40 but less than 50, run `blue`

If the percentage of light is less than 40, run `black`



```
to yellow
talkto "Color
cleartext
print [Yellow]
print "| |
print [OK to pack.]
end

to blue
talkto "Color
cleartext
print [Blue]
print "| |
print [OK to pack.]
end

to black
talkto "Color
cleartext
print [Black]
print "| |
print [Return to spray painting booth!]
end
```

Address the text box named `Color`
Erase any text in the text box
Print the word `yellow` **in the text box**
Print a carriage return (blank line)
Print the phrase in the text box

Address the text box named `Color`
Erase any text in the text box
Print the word `blue` **in the text box**
Print a carriage return (blank line)
Print the phrase in the text box

Address the text box named `Color`
Erase any text in the text box
Print the word `black` **in the text box**
Print a carriage return (blank line)
Print the phrase in the text box

▶ To operate the color code reader system, place an element in front of the light sensor. Then, click on the Check Color button.

The `check.color` procedure runs. It evaluates input from the light sensor, then runs the appropriate yellow, blue or black subprocedure. The subprocedure reports the color of the element. An OK message is printed if the element is yellow or blue. A "Return to spray painting booth" message is printed if the element is black.

Programming Ideas Example B

See *Introductory Explorations 1-4* in the *Setup Guide and Introductory Explorations* book for information about using the sound element and the light sensor.

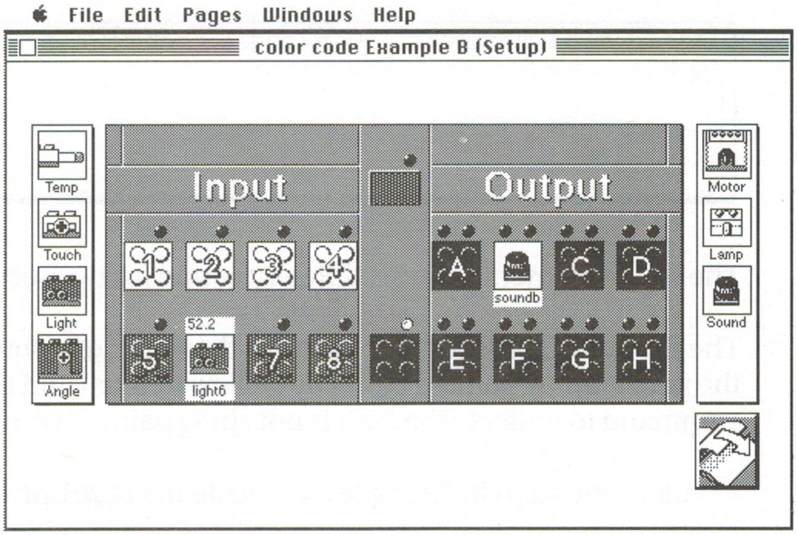
This example addresses the additional criteria in the student design brief. It shows how to add two sounds to the system: one indicating an acceptable item and another indicating an unacceptable item.

▶ Load LEGO DACTA® Control Lab software. Start a New Project or Open the code reader project with the Example A solution.

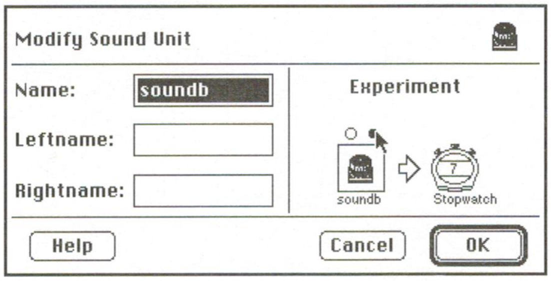
▶ Make sure a light sensor is attached to input port 6 on the LEGO DACTA Serial Interface and a light sensor icon is placed on input port 6 of the Setup page.

▶ Attach a sound element to output port B of the interface box.

▶ Drag a sound element icon to output port B on the Setup page.



▶ Double-click to open the `soundb` dialog box.


Modify Sound Unit

Name: soundb

Leftname:

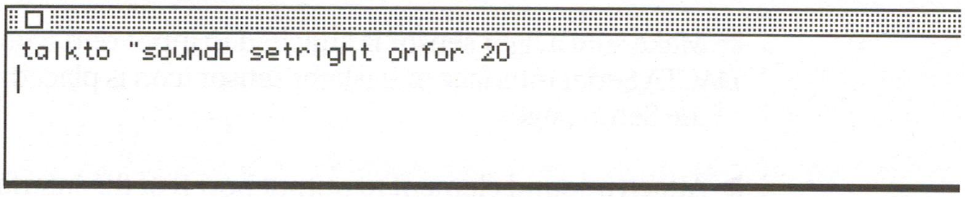
Rightname:

▶ Click on the right hand circle above the sound icon to hear the sound. Note the type of sound that is made.

If no sound is heard, turn the top of the sound element and try again.

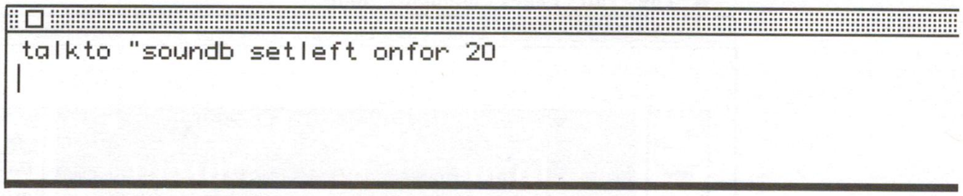
▶ Click on the left hand circle above the sound icon to hear the sound. A different type of sound is made.

▶ Program the sound element to turn on one type of sound for two seconds by typing the following in the Command Center.



```
talkto "soundb setright onfor 20
```

▶ Change the sound by changing the direction electricity flows through the sound element. Type the following:



```
talkto "soundb setleft onfor 20
```

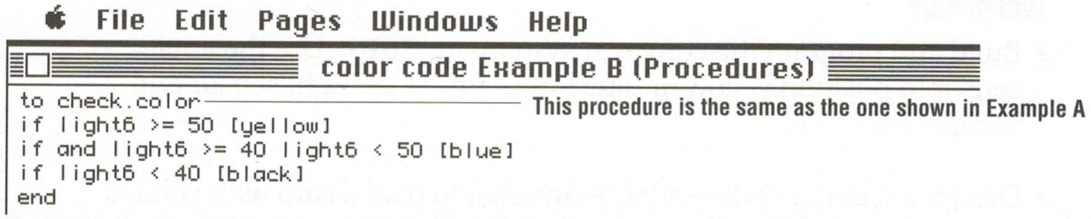
The sound element makes a different sound for two seconds.

The Example B procedures below use the `setright` sound to indicate the item is spray painted correctly (yellow or blue) and the `setleft` command to indicate the item is not spray painted correctly (black).

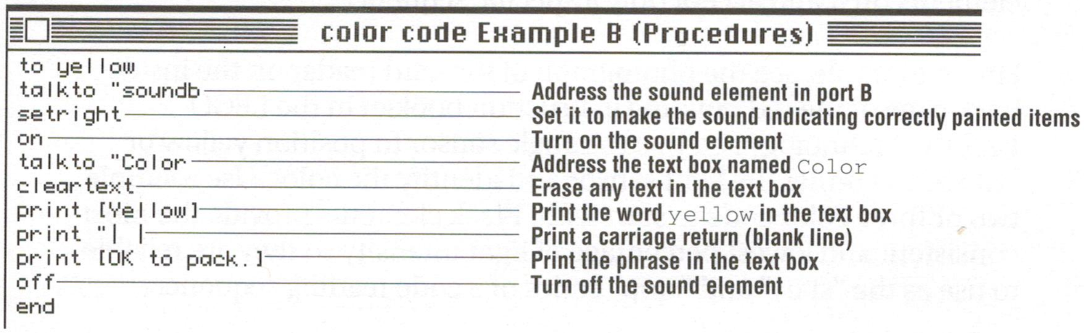
If you already have the Example A project programmed, you can move on to writing procedures because the project page button, monitor and text boxes require no changes from those shown in Example A.

▶ Follow the steps in Example A to create the Page1 project page.

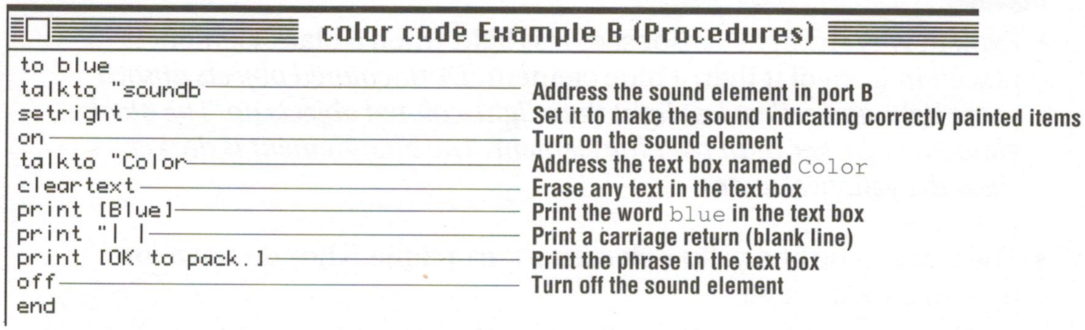
▶ After creating the project page elements, go to the Procedures page and enter the procedures and subprocedures as shown next.


File Edit Pages Windows Help
color code Example B (Procedures)

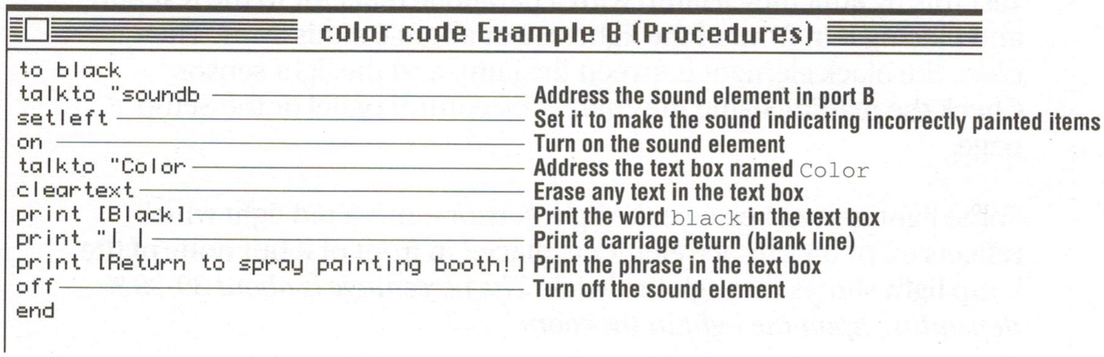
`to check.color`	This procedure is the same as the one shown in Example A
`if light6 >= 50 [yellow]`	
`if and light6 >= 40 light6 < 50 [blue]`	
`if light6 < 40 [black]`	
`end`	



`to yellow`	
`talkto "soundb`	Address the sound element in port B
`setright`	Set it to make the sound indicating correctly painted items
`on`	Turn on the sound element
`talkto "Color`	Address the text box named `Color`
`cleartext`	Erase any text in the text box
`print [Yellow]`	Print the word `yellow` in the text box
`print "| |`	Print a carriage return (blank line)
`print [OK to pack.]`	Print the phrase in the text box
`off`	Turn off the sound element
`end`	



`to blue`	
`talkto "soundb`	Address the sound element in port B
`setright`	Set it to make the sound indicating correctly painted items
`on`	Turn on the sound element
`talkto "Color`	Address the text box named `Color`
`cleartext`	Erase any text in the text box
`print [Blue]`	Print the word `blue` in the text box
`print "| |`	Print a carriage return (blank line)
`print [OK to pack.]`	Print the phrase in the text box
`off`	Turn off the sound element
`end`	



`to black`	
`talkto "soundb`	Address the sound element in port B
`setleft`	Set it to make the sound indicating incorrectly painted items
`on`	Turn on the sound element
`talkto "Color`	Address the text box named `Color`
`cleartext`	Erase any text in the text box
`print [Black]`	Print the word `black` in the text box
`print "| |`	Print a carriage return (blank line)
`print [Return to spray painting booth!]`	Print the phrase in the text box
`off`	Turn off the sound element
`end`	

▶ To operate the control panel, place an element in front of the light sensor. Then click the Check Color button.

The `check.color` procedure runs. It identifies the color of the element and runs the appropriate subprocedure. The subprocedure turns on the sound element, prints a message in the Color text box, then turns off the sound element.

`Cleartext` will erase any text, including the Procedures page. Students must make sure that they are addressing ("talking to") the correct text area before they type `cleartext` in the Command Center or run a procedure using `cleartext`.

Related Concepts and Extensions (Interdisciplinary)

Technology

- Build and program the conveyor system on 9701-6. Use the angle sensor to position yellow or blue bricks under the light sensor and identify the color.

- Design a security code reader. Program it to read a card with colored elements on it and accept only a specific sequence.

  For an example, see the photograph of the card reader on the inside back page of the red building instruction booklet in the LEGO DACTA® Technology set. Use the angle sensor to position yellow or blue bricks under the light sensor and identify the color. Use a simple two or three color code to start with. Black elements provide the most consistent and lowest percentage of light intensity so they are reliable to use as the "start" and "stop" codes of a code reading sequence.

Science

- Explain why the light sensor sees less light when a black element is placed in front of it than a blue element. *Dark-colored objects absorb more light and reflect less light than light-colored objects do. The black element is darker than the blue element. The blue element is darker than the yellow element.*

- The black, blue and yellow elements are **opaque**. They do not allow light to pass through.

  Test this by attaching a lamp with a parabolic reflector to the test port and placing it in front of the light sensor about 6 studs away. Then place the black element between the lamp and the light sensor. Check the reading using the color code control panel or the Setup page.

  Some light is seen because the light sensor emits a red light which reflects off of the surface elements placed in front of it but none of the lamp light shines through the brick. *The percentage is about 30-39 % depending upon the light in the room.*

  The percentage figures given will vary depending on the light in the room.

  Take the opaque bricks away so the lamp is shining directly into the light sensor. Check the reading of the light. *The percentage is about 100%.*

  Transparent red, yellow, blue and green elements are included in the LEGO DACTA® Technology set. Transparent matter allows light to pass through.

Test the transparent bricks by placing a stack of transparent elements between the lamp and the light sensor. Check the light sensor feedback. *The percentage is about 98%.*

Math

- Use a protractor and a mirror along with a Control Lab lamp and light sensor to measure the amount of light from a lamp (**incident ray**) after it is reflected (**reflected ray**). Place a lamp on the base plate facing one direction. Use the mirror to reflect light from the lamp and direct it at different angles. Measure the light intensity reflected into the light sensor. For each position of the mirror, find the highest reading possible from the light sensor. Draw rays from the lamp to the mirror and from the mirror to the light sensor to show what you think happens to light as it reflects off of the mirror. *The angle of incidence (light from the lamp) is equal to the angle of reflection.*

Language Arts

- Communication systems can be used to inform, educate, persuade and entertain. Using the solutions shown in Examples A and B, the color code system informs the operator whether an item has been correctly spray painted or not.

  Change the system to satisfy another purpose for communication:
  —Add a page to the color code reader system to persuade someone to purchase it.
  —Add a page that instructs someone how to use it.
  —Entertain people by redesigning the system so the yellow, blue and black elements sound a simple series of tones to make up a song.

Social Studies

- You are the person in charge of running fair elections for a small, remote town. Unfortunately, many people do not read well. In order to encourage people to vote, you have assigned each of the three candidates a color. People to vote for the candidate of their choice by selecting one of the yellow, blue or black elements and placing it in front of the light sensor code reader.

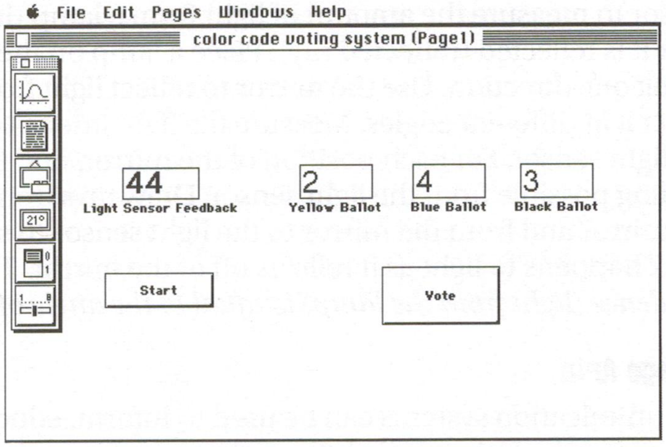
  Redesign the color code system to set up three variables that count up the votes for each color. Add one to the appropriate variable each time the color is placed in front of the light sensor. Students can use monitor boxes to show the current value of the variable.

Students can also use text boxes to print the value of the variable. For example, if yellow is a text box, type:

```
talkto "|Yellow|
cleartext
print :yel
```

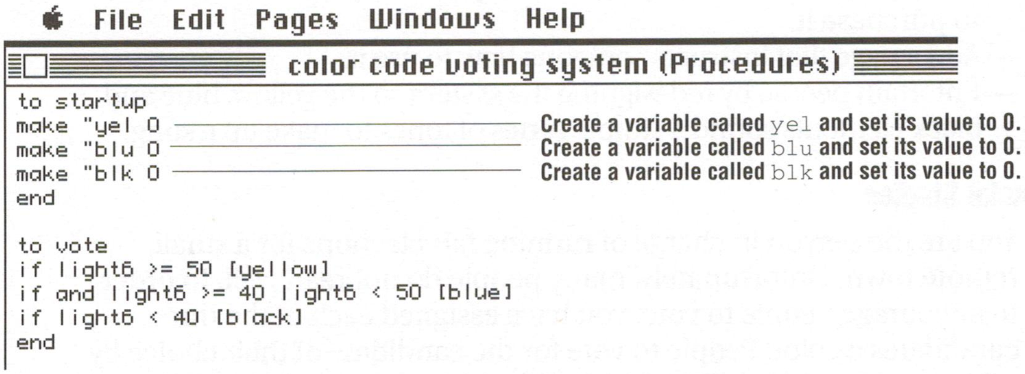
Below is an example of what the tally system for the election could look like. The same light sensor, sound element, and yellow, blue and black bricks are used as they were in the color code system. The check color button and `check.color` procedure have both been renamed Vote.

See the `tone` command in the *Reference Guide.*


File Edit Pages Windows Help
color code voting system (Page1)
Start
Vote

44	2	4	3
Light Sensor Feedback	Yellow Ballot	Blue Ballot	Black Ballot

Click the Start button only once for each election. The `startup` procedure sets all three color variables to zero. Each time someone "votes," he or she must place an element in front of the light sensor and click the Vote button.

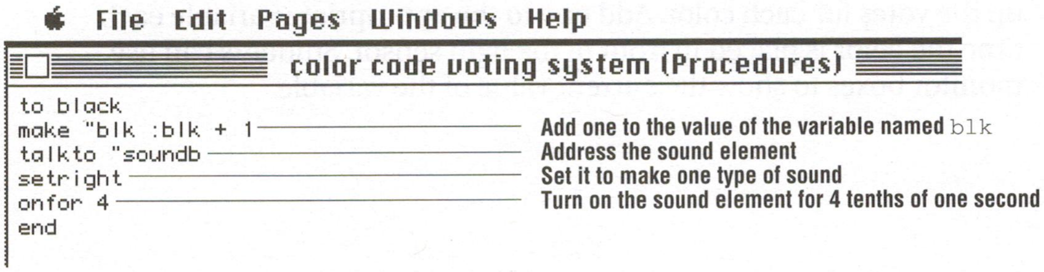

File Edit Pages Windows Help
color code voting system (Procedures)

```
to startup
make "yel 0
make "blu 0
make "blk 0
end

to vote
if light6 >= 50 [yellow]
if and light6 >= 40 light6 < 50 [blue]
if light6 < 40 [black]
end
```

Create a variable called `yel` and set its value to 0.
Create a variable called `blu` and set its value to 0.
Create a variable called `blk` and set its value to 0.

The subprocedures keep track of the number of votes for each color. Write a subprocedure like the `black` subprocedure for the yellow and blue colors.


File Edit Pages Windows Help
color code voting system (Procedures)

```
to black
make "blk :blk + 1
talkto "soundb
setright
onfor 4
end
```

Add one to the value of the variable named `blk`
Address the sound element
Set it to make one type of sound
Turn on the sound element for 4 tenths of one second

Bridge: A Production/Construction Invention

Student Name: ______________________________ **Date:** ________________

Background Information

What is Production / Construction technology?

Production involves changing resources into materials that can be purchased by consumers. For example, raw materials such as wood can be milled and refined into other usable materials such as lumber. Construction is a specific type of Production. Construction projects involve building homes, roads, tunnels and bridges.

How are bridges built?

A bridge can be built from wood, stone, concrete, steel or other material depending upon the type of traffic it carries, the width of the span it must cross, the nature of the river bed or earth upon which the bridge foundation is built, and the capital available.

How can bridges be made to move?

Bridges over small waterways are often built low, so they must be designed to allow water traffic to pass.

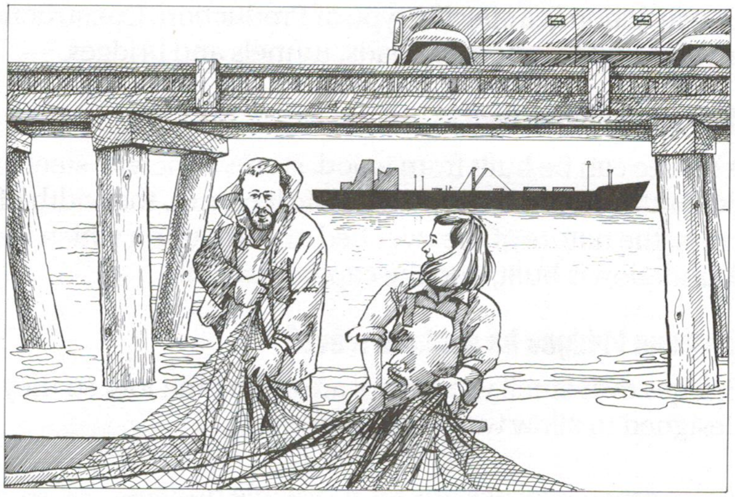
There are several designs for moveable bridges.

Swing Span	A swing span bridge has a section mounted on a turntable. To open, the turntable rotates the span 90° or one quarter of a turn so boats can pass on either side.
Drawbridge	A drawbridge is a simple wooden span that can be raised and lowered using chains. Medieval castles in Europe were designed with drawbridges as security to protect castle entrances.
Bascule	A modern concrete and steel version of the drawbridge design, a bascule bridge can be raised and lowered using concrete weights and motors.
Vertical Lift	A vertical lift bridge includes a section that moves up and down. Counterweighted cable and pulley systems slide up and down in towers that look like elevators on either end of the moveable span. Compared to other types of moveable bridges, vertical lift bridges open farther to allow larger ships to pass.

How are moveable bridges controlled?

Moveable bridges must have a system for opening and closing the bridge safely. Usually a bridge operator or *bridgetender* monitors water and road traffic from a control house on the bridge. The bridgetender is responsible for opening and closing the bridge at the appropriate times.

Define The Problem



Problem Statement

Many jobs can be created in Jacksontown if a new bridge that accommodates larger ships can be built to span the river.

Design Brief

▶ Design and assemble a bridge that moves. Program a control system to operate the bridge.

The bridge design must include the following:

- A motorized system that lifts to allow large ships to pass and lowers to allow vehicles to cross the bridge.
- Easily operated computer controls to lift and lower the bridge.

Additional Criteria

The bridge system can also include:

- A computer-controlled slider that adjusts the motor speed.
- A feedback system that monitors the bridge movement.
- A computer-control that halts the motors and warns the bridgetender that the bridge needs adjusting.
- An emergency stop button that halts the motors if the bridgetender notices a problem.

Building Ideas

▶ Design a vertical lift bridge that includes two motor and gear subsystems to raise and lower the bridge.

- The two motor and gear subsystems must have identical gear reduction.
- If pulleys are used for part of the gear reduction, they should be at the beginning of the subsystem near the motor and should be followed by gears. Gears allow more precise movement.
- If string is used for cables, attach it securely by threading the string through a bushing. Then push the bushing onto an axle. String can also be fastened to beams using connector pegs.
- Build up the sides of the bridge for structural stability. The bridge must stay together as it moves. The sides can also serve as safety features for the automobiles crossing the bridge.
- Use tires as bearings to guide the movement along any large beams.

▶ Include the angle sensors in the two motor subsystems if you want feedback on the bridge movement.

The angle sensor feedback can be used to compare the position of each side of the bridge and ensure the two sides stay aligned.

Programming Ideas

▶ Load LEGO DACTA® Control Lab software. Start a New Project.

▶ Use the Setup Page and the Command Center to test the motorized subsystems of the bridge model to make sure the bridge is moving properly. Adjust the tautness of the string, the direction it winds and the motor speed and direction to correct any misalignment.

Use the stop button on the LEGO DACTA® Serial Interface to halt the motors. Press the stop button again to resume.

The two motors must be addressed at the same time so the movement is synchronized. To address both motors at the same time, put the motor names in square brackets after the `talkto` command.

```
talkto [motora motorb]
```

See the *Quick Start Guide* section "Using the Angle Sensor" for background information on using one angle sensor.

▶ If you are using the angle sensors for feedback, you must find out:

- At which point is the bridge low enough so the angle sensors can be reset?
- At which rotational count for each sensor is the bridge up high enough?

▶ Create a control panel to operate the bridge. Review the design brief specifications so that you include the important features.

▶ Define the buttons controlling the motors so that they turn on when the mouse button is pressed and turn off when the mouse button is released. Or, add an emergency stop button to the control panel so that both motors can be stopped immediately if a problem arises.

▶ Use the procedures below as part of the control system to monitor the bridge movement and sound a warning when the bridge becomes misaligned.

To synchronize the motors with angle sensor feedback, input from both sensors must be compared to see if the difference between the angle sensor feedback is within a specified range.

The `compare` procedure below finds the difference between the two angle sensors and reports the number to the `check` procedure. The command `abs` (absolute value) allows you to compare angle sensor values regardless of whether they are positive or negative. `Check` evaluates whether the difference between the sensors is above 8 (you can use a higher or lower number if you wish) or not. If the difference is higher than 8, the motors turn off, a tone sounds and the bridge procedure stops before any greater misalignment occurs.

Absolute values are always positive or zero. Find absolute values by leaving positive numbers alone and ignoring the sign of negative numbers.

```
to check
if compare > 8 [alloff tone 72 1 stopall]
end

to compare
output abs (angle5 - angle6)
end
```

Teacher Notes for Bridge: A Production/Construction Invention

✏ **Teacher note:**

See the "What Do Students Learn?" section in the beginning of this book for concepts in technology, science, math, language arts and social studies that are addressed by the Bridge activity.

Objectives

The purpose of the Bridge invention is to introduce students to construction systems through designing, building and programming a moveable bridge. By the end of the project, students will be able to:

1. Describe the process they used to develop the bridge.
2. Explain why bridges are examples of construction.

Students will also develop specific skills working with the LEGO DACTA® Control Lab output and input devices and control features. By the end of the project, students will be able to:

1. Design and build a motorized bridge.
2. Use pulley and gear ratios to reduce the speed of the motor.
3. Use software tools and commands to develop a control panel for the bridge.
4. Control the bridge using the Control Lab software and a control panel.

Materials Needed

- One LEGO DACTA Technology Set and building instructions for each group (2 - 3 students).
- At least one Control Lab system.
- Photocopies of the student Bridge pages.
- Disks or disk storage space for each student group project.
- String, if students build the solution shown in the Building Ideas of these Teacher Notes.

Background Information

To prepare students for the bridge activity, two pages of background information provide answers to the following questions: What is Production / Construction technology? How are bridges built? How can bridges be made to move? How can I control a moveable bridge?

See the Bibliography in the Supplemental Materials section of this book for additional references.

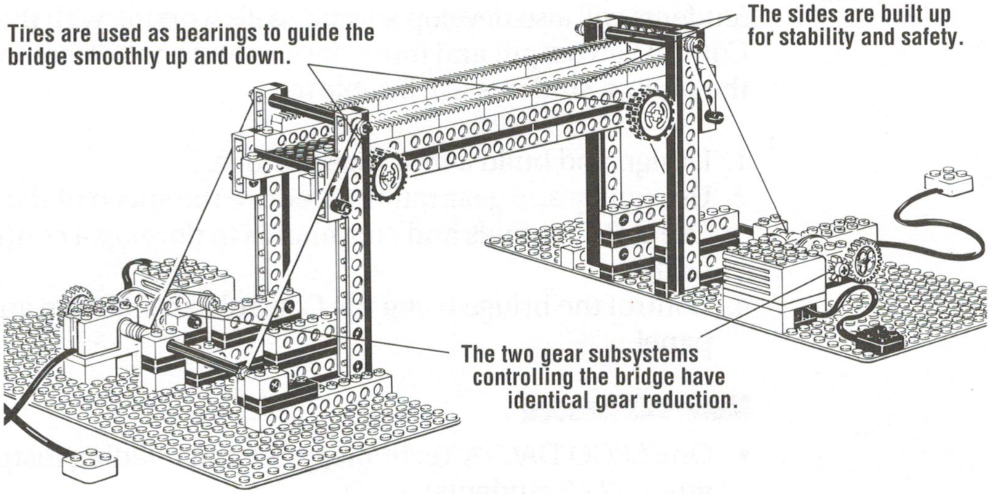
Define the Problem

Several criteria are listed on the student design brief page. You may wish to add your own.

Building Ideas

Students can build a model similar to the one shown below or they can design and build their own bridge. The bridge activity is an "Invention" which means no step-by-step instructions are given.

When adjusting the string, fit the string through a bushing, then mount the bushing on an axle. That way, the string will hold firm as the bridge moves. Next, hook the string over the pulley on top of the bridge. Finally, fit the other end of the string into the grooves of a connector peg and lock the connector peg into place using a beam hole. Remember to keep tension on the string throughout this final step. The direction the string winds around the axle also affects the direction the bridge raises and lowers.



Tires are used as bearings to guide the bridge smoothly up and down.

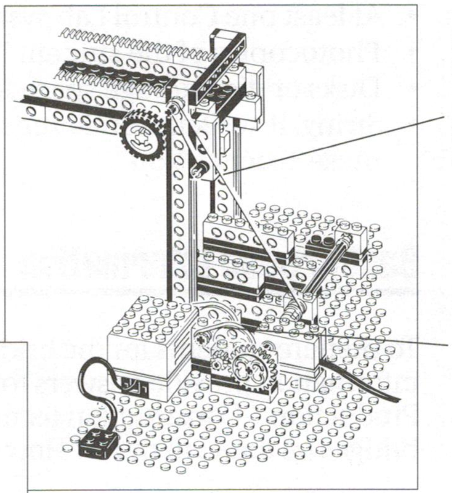
The sides are built up for stability and safety.

The two gear subsystems controlling the bridge have identical gear reduction.

Mat board can be used to create a smooth "driving surface" for any vehicles crossing the bridge.

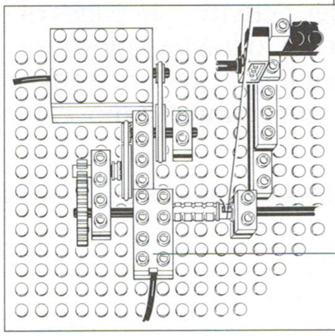
Add beams across to connect the two baseplates, if the model needs more stability.

The bridge is 6 studs across: four are visible when the bridge sides are connected. Use the black plates with and without holes for the "road."



The strings should be fastened to the axles using bushings. Connector pegs can be used to fasten the string to the beams.

Pulleys are used for part of the gear subsystem but they are mounted near the motor and are followed by gears to reduce the impact of slippage.



An angle sensor is mounted in the drive train of both sides of the bridge to provide feedback for aligning the movement of the bridge.

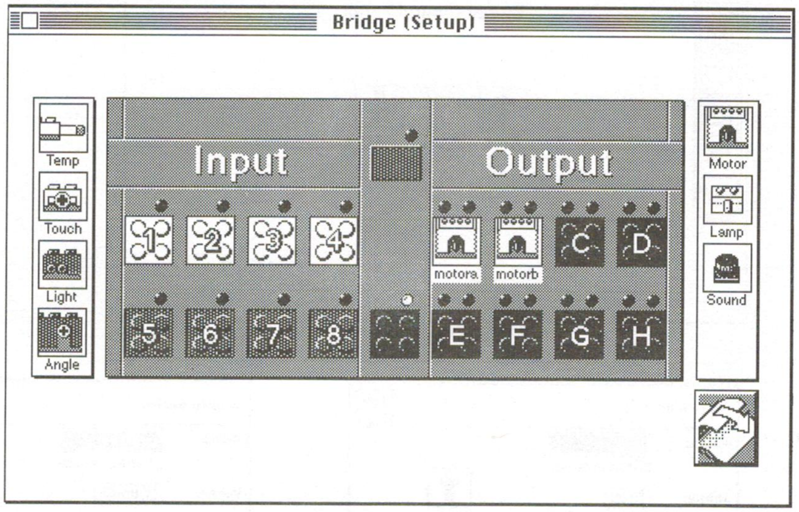
Programming Ideas: Example A

This example addresses the main criteria in the student design brief. It shows a control panel with two buttons for raising and lowering the bridge.

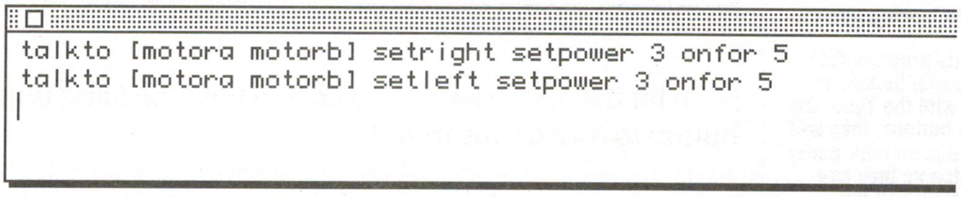
▶ Load LEGO DACTA® Control Lab software. Start a New Project.

▶ Attach one motor to output port A and the other to output port B on the LEGO DACTA® Serial Interface.

▶ Drag motor icons to ports A and B on the Setup page.



▶ Make sure the motors are running at the appropriate speed by typing the following in the Command Center.



```
talkto [motora motorb] setright setpower 3 onfor 5
talkto [motora motorb] setleft setpower 3 onfor 5
```

If two motors are used, one for each side of the bridge, the motors must be addressed at the same time so the movement is synchronized. In Logo, when more than one thing is grouped together, it is put into a "list." A list is a sequence of words or other lists which are surrounded by square brackets.

Use the stop button on the interface box to halt the motors in an emergency. Press the stop button again to resume.

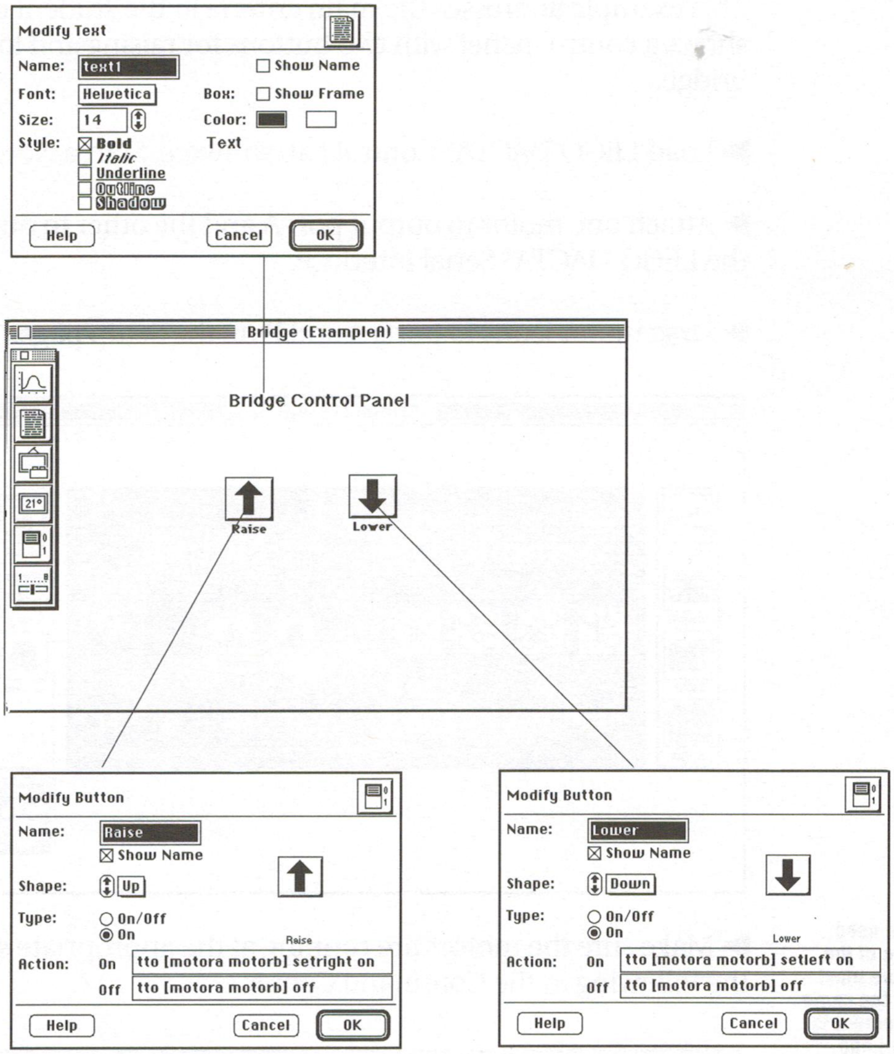
▶ If the bridge is not moving smoothly up and down, try the following:

- make sure the string for the "cables" are the same length, the same tautness and wound the same way around the axle.
- make sure the motors are moving the same direction (flip the connector on the output port to reverse the direction).
- adjust the bushings at the top of the beams so the friction is not interfering with the moving part of the bridge.

▶ Create a control panel to operate the bridge.

▶ Use the button tool to make buttons for raising and lowering the bridge.

For more information, see the *Quick Start Guide* section, "Using the Text Tool."



Modify Text

Name: text1 ☐ Show Name

Font: Helvetica Box: ☐ Show Frame

Size: 14 Color:

Style: ☒ Bold ☐ Italic ☐ Underline ☐ Outline ☐ Shadow

Text

Bridge (ExampleA)

Bridge Control Panel

Modify Button

Name: Raise

☒ Show Name

Shape: Up

Type: ○ On/Off ● On

Action: On tto [motora motorb] setright on

Off tto [motora motorb] off

Modify Button

Name: Lower

☒ Show Name

Shape: Down

Type: ○ On/Off ● On

Action: On tto [motora motorb] setleft on

Off tto [motora motorb] off

See the *Quick Start Guide* sections "Using the Button Tool" and Changing the Type of Button" for more information.

With this Type: On button, an emergency stop button is not needed because the student is always in control of the moving motors.

If students program their control panel buttons to operate with the Type: On/Off style buttons, they will need to add an emergency stop button so they can immediately halt the motors if a problem arises.

▶ To lift the bridge section, click and hold the Raise button. Release the button to turn off the motor.

▶ Use the text tool to make a heading for the control panel.

When new pages are created, they are automatically named in sequence, e.g. Page1, Page2. Pages can be named other names. For example, in the Programming Ideas the Example A solution the page has been renamed ExampleA. To rename a page ExampleA, select the page from the Pages menu so it is showing on your screen. Then type in the Command Center:

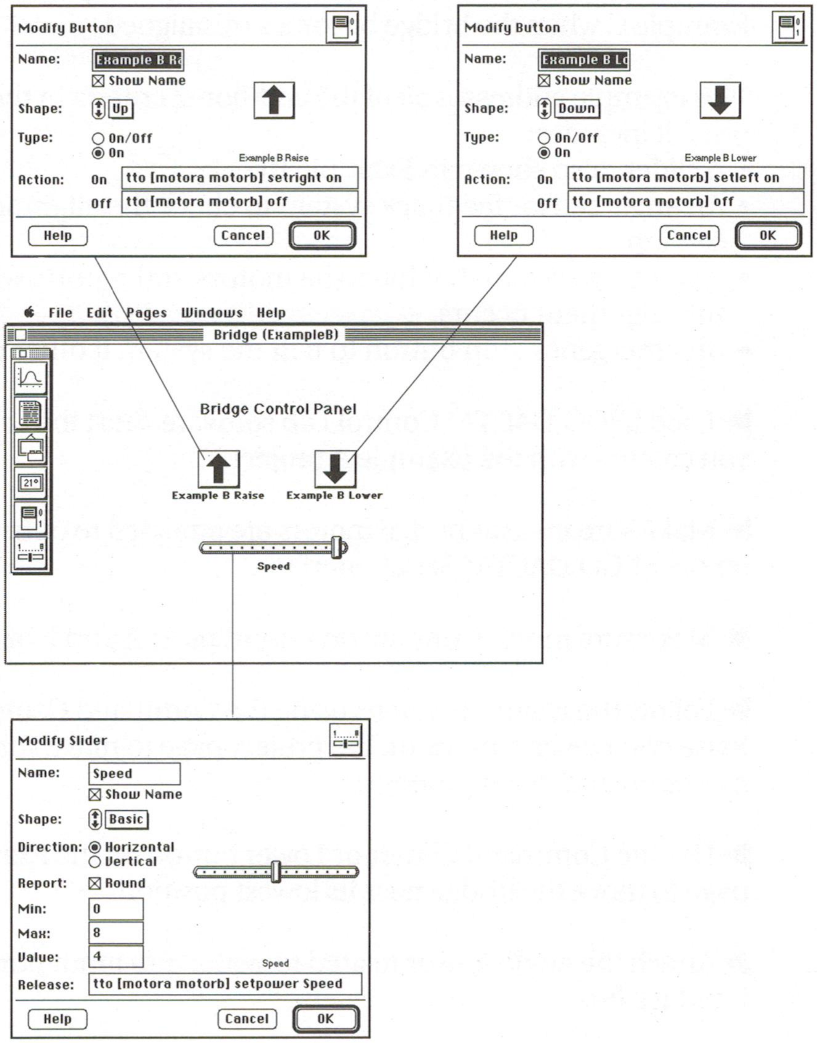
See the *Reference Guide* for more information on the `namepage` command.

```
namepage "ExampleA
```

Programming Ideas: Example B

See the *Quick Start Guide* "Using the Slider Tool" for more information.

This example addresses one of the additional criteria: adding a slider to the control panel for changing the power level of the motor.



Modify Button
Name: Example B Ra
☒ Show Name
Shape: Up
Type: ○ On/Off ● On
Action: On tto [motora motorb] setright on
Off tto [motora motorb] off

Modify Button
Name: Example B Lo
☒ Show Name
Shape: Down
Type: ○ On/Off ● On
Action: On tto [motora motorb] setleft on
Off tto [motora motorb] off

Modify Slider
Name: Speed
☒ Show Name
Shape: Basic
Direction: ● Horizontal ○ Vertical
Report: ☒ Round
Min: 0
Max: 8
Value: 4
Release: tto [motora motorb] setpower Speed

If you are adding Example B as a separate page to the project, rather than just adding a slider to Example A, follow these steps.

- Shift-click on the Raise and Lower buttons to select both of them.
- Select Copy from the Edit menu to copy them.
- Select New Page from the Pages menu to create a new page.
- Select Paste from the Edit menu to paste the buttons onto the new page.
- Double-click each button and change the names to Example B Raise and Example B Lower.

You cannot duplicate page element names within the same project so you cannot call the buttons on the new page Raise and Lower.

It is not necessary to try Example B.

Example C

Complete Example A before trying Example C. You will need to use the Raise and Lower buttons on Example A to correct problems with Example C when the bridge becomes misaligned.

This example addresses all of the additional criteria in the student design brief. It includes:
- A slider (also shown in Example B).
- An angle sensor feedback system to check the alignment of the bridge section.
- A warning system that halts the motors and reports when a bridge misalignment occurs.
- An emergency stop button to halt the system if other problems occur.

▶ Load LEGO DACTA® Control Lab software. Start the bridge project file you created with the Example A project.

▶ Make sure the two bridge motors are attached to output ports A and B on the LEGO DACTA® Serial Interface.

▶ Make sure motor icons are on output ports A and B of the Setup page.

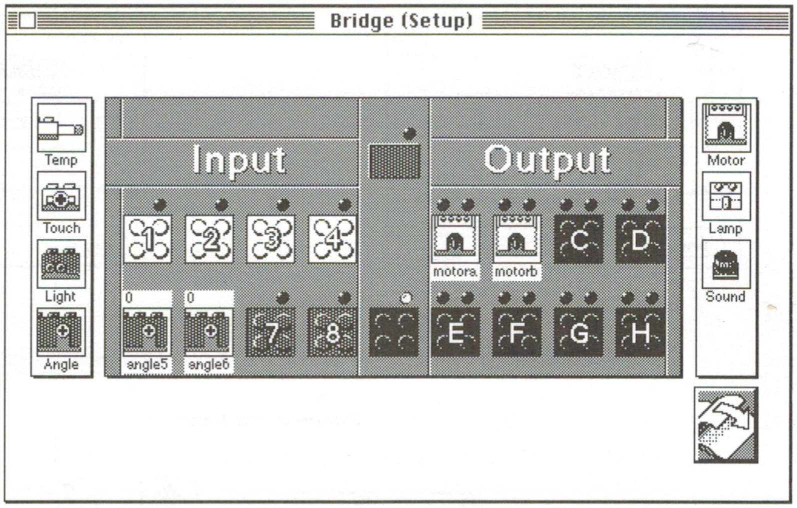
▶ Follow the Example A steps using the Command Center or click on the Raise and Lower buttons on the project page to make sure the bridge moves up and down properly.

▶ Use the Command Center or Lower button on the Example A project page to move the bridge near its lowest position.

▶ Attach the angle sensor related to `motora` to input port 5 on the interface box.

▶ Attach the angle sensor related to `motorb` to input port 6 on the interface box.

▶ Then on the Setup page, drag angle sensor icons to input ports 5 and 6.



▶ Reset the angle sensors to zero by typing the following in the Command Center.

```
resetrotation 5 resetrotation 6
```

Example C uses 4 for both sensors in the `down` procedure.

Use a number slightly above 0 when programming the `down` procedure.

▶ Then type the following in the Command Center.

```
talkto [motora motorb] on
```

▶ Press the stop button on the LEGO DACTA® Serial Interface to halt the motors when the bridge approaches the top.

▶ Write down the angle sensor values.

Example C uses 25 as the number for both sensors in the `up` procedure.

These are the numbers to use when programming the `up` procedure.

▶ Type `off` in the Command Center and then press the stop button on the interface box again to resume power to the output ports.

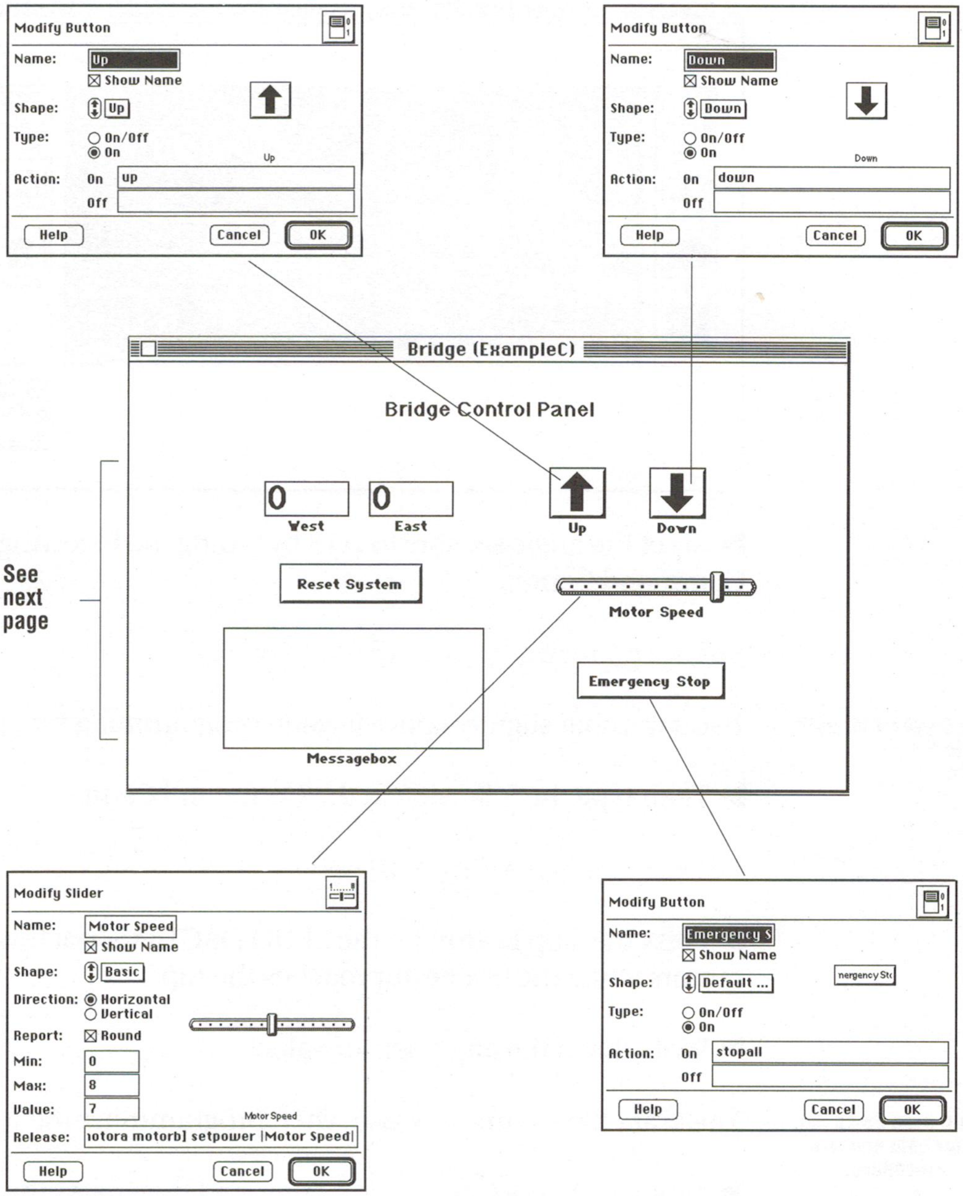
▶ Create a control panel to operate the vertical lift bridge.

Example C includes the following features:
- Up and down buttons to lift and lower the bridge.
- A slider to control the motor speed.
- Two monitors to show the angle sensor feedback.
- A reset button to reset the angle sensors to 0.
- An emergency stop button to halt the motors.
- A text box for a warning message if the bridge is misaligned.

Notice in the dialog boxes for the buttons that procedure names are typed in the Action: On area. That means the procedure is executed when the button is activated.

All procedures are typed and stored on the Procedures page. LEGO DACTA® Control Lab software finds the procedure on the Procedures page and executes it. The procedure name used in a dialog box must match the procedure name on the Procedures page exactly, or Control Lab responds with an error message.

Writing Logo statements on the Procedures page makes writing and editing them much easier, especially if they are more than two or three commands long.



Name: Up
☒ Show Name
Shape: Up
Type: ○ On/Off ● On
Action: On up
Off

Name: Down
☒ Show Name
Shape: Down
Type: ○ On/Off ● On
Action: On down
Off

Name: Motor Speed
☒ Show Name
Shape: Basic
Direction: ● Horizontal ○ Vertical
Report: ☒ Round
Min: 0
Max: 8
Value: 7
Release: ıotora motorb] setpower |Motor Speed|

Name: Emergency S
☒ Show Name
Shape: Default ...
Type: ○ On/Off ● On
Action: On stopall
Off

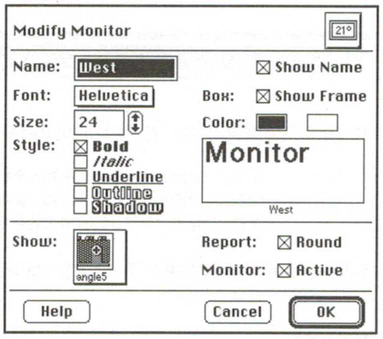
See next page

If you have Example B and Example C control panels in the same project, you must name the slider on the Example C page something other than Speed, since that name was used for the slider on the Example B page. Motor Speed is the name shown on Example C. Vertical bars | | tell Control Lab to treat two separate words as one word. So the Release line in the dialog box of a slider named Motor Speed would be:

```
talkto [motora motorb] setpower |Motor Speed|
```

The Emergency Stop button halts both motors and stops all `forever` procedures that are running.

West angle sensor monitor dialog box.



Modify Monitor

Name: West ☒ Show Name

Font: Helvetica Box: ☒ Show Frame

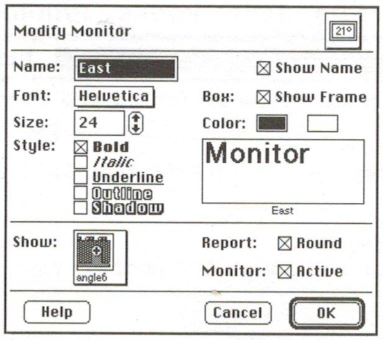
Size: 24 Color:

Style: ☒ Bold ☐ *Italic* ☐ Underline ☐ Outline ☐ Shadow

Show: angle5 Report: ☒ Round

Monitor: ☒ Active

East angle sensor monitor dialog box.



Modify Monitor

Name: East ☒ Show Name

Font: Helvetica Box: ☒ Show Frame

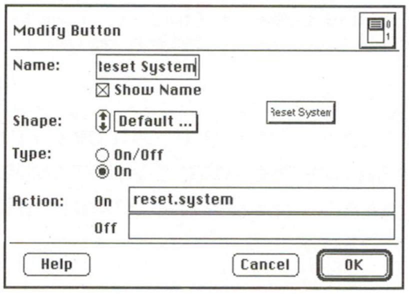
Size: 24 Color:

Style: ☒ Bold ☐ *Italic* ☐ Underline ☐ Outline ☐ Shadow

Show: angle6 Report: ☒ Round

Monitor: ☒ Active

Reset System button dialog box.



Modify Button

Name: ieset System

☒ Show Name

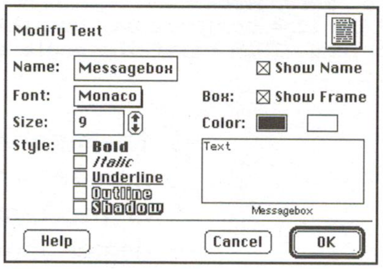
Shape: Default ...

Type: ○ On/Off ● On

Action: On reset.system

Off

Message box text dialog box



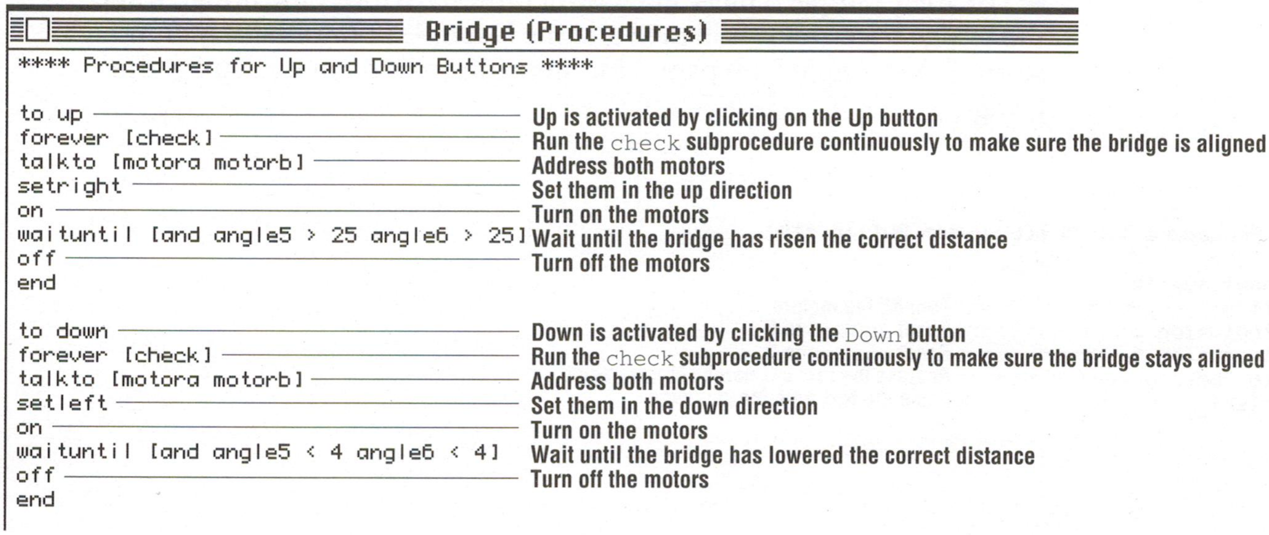
Modify Text

Name: Messagebox ☒ Show Name

Font: Monaco Box: ☒ Show Frame

Size: 9 Color:

Style: ☐ Bold ☐ *Italic* ☐ Underline ☐ Outline ☐ Shadow



Bridge (Procedures)

```
**** Procedures for Up and Down Buttons ****

to up
forever [check]
talkto [motora motorb]
setright
on
waituntil [and angle5 > 25 angle6 > 25]
off
end

to down
forever [check]
talkto [motora motorb]
setleft
on
waituntil [and angle5 < 4 angle6 < 4]
off
end
```

Up is activated by clicking on the Up button
Run the `check` **subprocedure continuously to make sure the bridge is aligned**
Address both motors
Set them in the up direction
Turn on the motors
Wait until the bridge has risen the correct distance
Turn off the motors

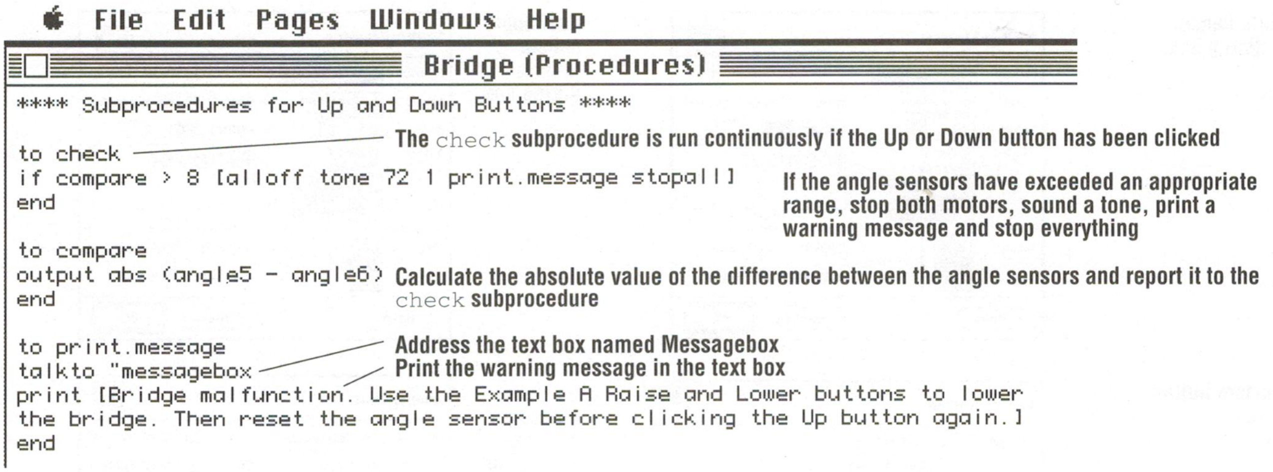
Down is activated by clicking the `Down` **button**
Run the `check` **subprocedure continuously to make sure the bridge stays aligned**
Address both motors
Set them in the down direction
Turn on the motors
Wait until the bridge has lowered the correct distance
Turn off the motors

You may need to use a number other than 25. See the beginning of Example C for steps to figure out the correct number.

You may need to use a number other than 4. See the beginning of Example C for steps to figure out

If the angle sensor feedback is used to synchronize the motor, input from both sensors must be compared to see if the difference between the angle sensor feedback is within a specified range.

The `check` and `compare` procedures are given on the student Building and Programming Ideas pages. The `compare` procedure finds the difference between the two angle sensors and reports the number to the `check` procedure. `Check` evaluates whether the difference between the sensors is above 8 (you can use a higher or lower number if you wish) or not. If the difference is higher than 8, the motors turn off, a tone sounds and the bridge system stops before any greater misalignment occurs.



```
**** Subprocedures for Up and Down Buttons ****

to check
if compare > 8 [alloff tone 72 1 print.message stopall]
end

to compare
output abs (angle5 - angle6)
end

to print.message
talkto "messagebox
print [Bridge malfunction. Use the Example A Raise and Lower buttons to lower
the bridge. Then reset the angle sensor before clicking the Up button again.]
end
```

The `check` subprocedure is run continuously if the Up or Down button has been clicked

If the angle sensors have exceeded an appropriate range, stop both motors, sound a tone, print a warning message and stop everything

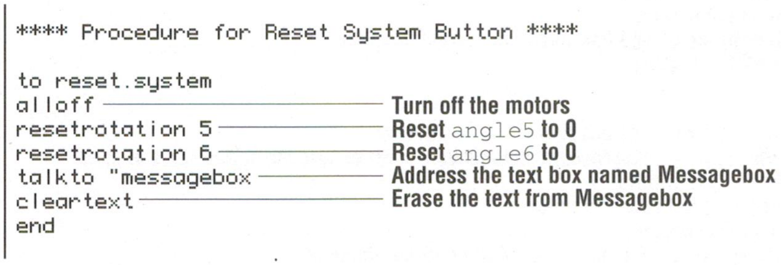
Calculate the absolute value of the difference between the angle sensors and report it to the `check` subprocedure

Address the text box named Messagebox

Print the warning message in the text box

The number 8 used in the `check` procedure can be changed to another number. A higher number means the system will tolerate a higher degree of misalignment. A lower number means the system will tolerate a lesser degree of misalignment.

▶ Use the Example A Raise and Lower buttons to lower the bridge when the bridge becomes misaligned and the Example C automatic system stops. See Example A on page 3 for more information on realigning the bridge.



```
**** Procedure for Reset System Button ****

to reset.system
alloff
resetrotation 5
resetrotation 6
talkto "messagebox
cleartext
end
```

Turn off the motors

Reset `angle5` to 0

Reset `angle6` to 0

Address the text box named Messagebox

Erase the text from Messagebox

Important:

`Cleartext` will erase any text, including the Procedures page. Students must make sure that they are addressing (“talking to”) the correct text area before they type `cleartext` in the Command Center or run a procedure using `cleartext`.

Related Concepts and Extensions (Interdisciplinary)

Technology

- Challenge students to add a warning system to sound when a ship approaches. The light sensor and a lamp could be used as an electric eye. When a ship passes by and breaks the beam of light, a sound element could be programmed to make a siren noise and warn that the bridge will be moving up to let the ship pass underneath.

Science

- Bridges expand in the heat and contract in the cold. How are these facts accommodated in bridge construction? *Overlapped teeth are used in the bridge joints to allow for expansion and contraction.*

Math

LEGO DACTA® Control Lab has four timers which measure in tenths of seconds. See the timer and resett commands in the *Reference Guide* for more specific information and examples.

- Time how long it takes to raise and lower the bridge. Then time how long it takes for a car to cross the bridge. How many cars will be backed up when the bridge must raise and lower?

  Design a traffic control system to prevent cars from crossing when the ship passes underneath. The wait for the cars should not be more than 20 seconds.

Language Arts

- Prepare a presentation of the bridge design to the Jacksontown Authority which must approve the project.

- Write a job description for the paper advertising the need for a bridge operator. Include ideas on the skills and experience needed to qualify for the job.

Social Studies

- Jacksontown citizens are willing to pay for the bridge because they believe it will create jobs and improve the local economy. However, developing the bridge does have impacts on other aspects of town life, such as the following:
  - degradation of the shellfish population due to potential oil spills and trash from increased ship traffic
  - lack of warehouse space on the dock
  - unsafe docks that must be repaired
  - job training for new workers

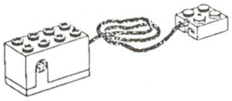
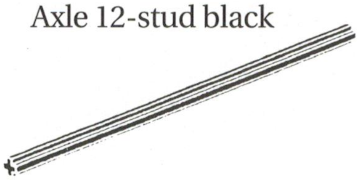
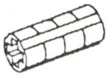
  You are a city council member who must devise a plan to handle these concerns. Prepare a list of new ordinances that can protect the environment and the dock workers. Also develop some incentives for property owners and business owners. Present your ideas at the next town meeting.

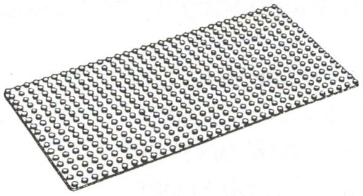
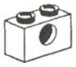
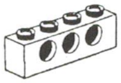
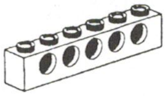
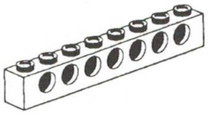
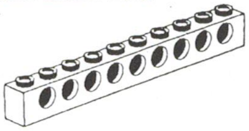
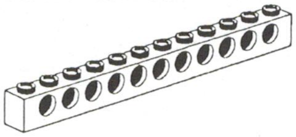
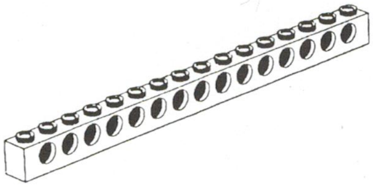
Supplemental Material

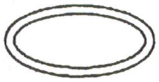
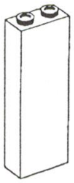
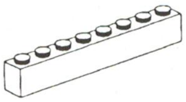
The materials in this section are optional copymasters provided to help students manage their projects. Some pages are specific to certain sections and are mentioned in the teacher notes.

6.2 Technology Set Parts List
6.6 Quick Reference Sheets
6.10 Measurement Worksheet
6.11 Front View Graph Paper
6.13 Top View Graph Paper
6.15 Problem Solving Steps for the Conceptual Framework Approach
6.17 Project Evaluation Checklist
6.19 Systems Problem Solving Flowchart
6.20 Systems flowcharts
  6.20 Open
  6.20 Program
  6.21 Closed
6.22 Bibliography

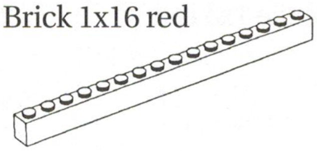
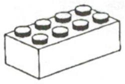
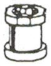
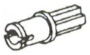
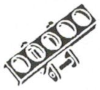
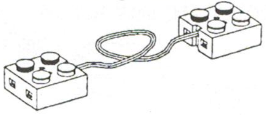
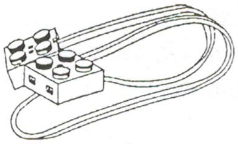
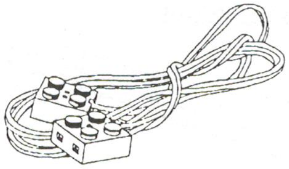
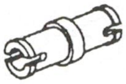
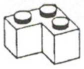
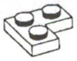
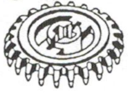
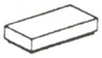
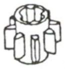
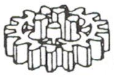
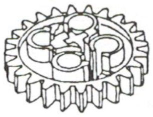
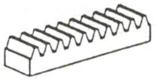
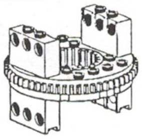
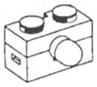
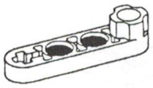
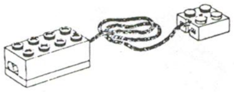
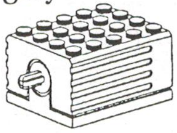
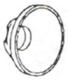
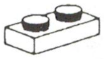
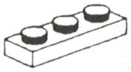
#9701 Technology Set Parts List

Part Name	Quantity	
Angle sensor 9v	2	
Angular brick 1x1 black	6	
Axle 3-stud black	2	
Axle 4-stud black	4	
Axle 6-stud black	6	
Axle 8-stud black	6	
Axle 10-stud black	2	
Axle 12-stud black	4	
Axle extender gray	2	

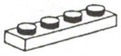
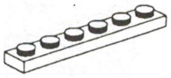
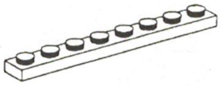
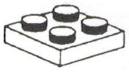
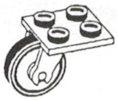
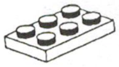
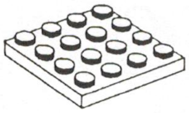
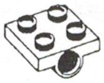
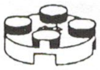
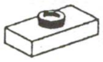
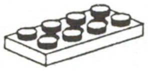
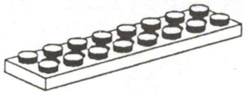
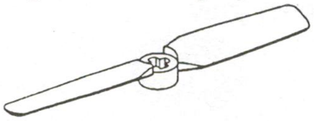
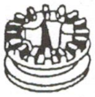
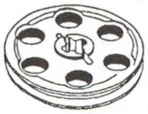
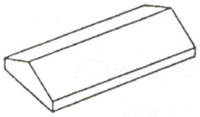
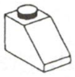
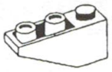
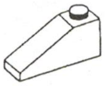
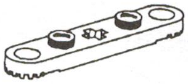
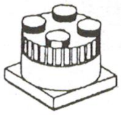
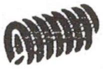
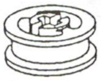
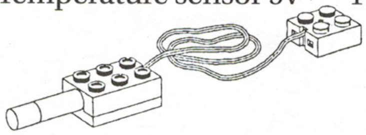
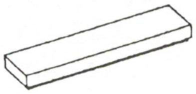
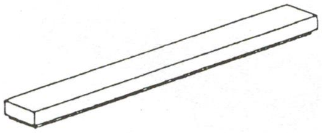
Part Name	Quantity	
Baseplate 16x32 gray	2	
Beam 1x2 red	16	
Beam 1x4 red	10	
Beam 1x6 red	10	
Beam 1x8 red	10	
Beam 1x10 red	8	
Beam 1x12 red	6	
Beam 1x16 red	6	

Part Name	Quantity	
Belt *(two 2.1 cm, two 2.5 cm, four 2.9 cm)*	8	
Brick 1x1 red	8	
Brick 1x2 *(ten black, fourteen red)*	24	
Brick 1x2x5 *(two blue, four yellow)*	6	
Brick 1x4 red	6	
Brick 1x6 *(four blue, four yellow)*	8	
Brick 1x8 red	2	

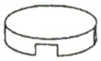
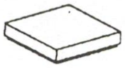
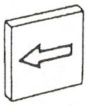
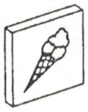
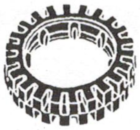
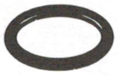
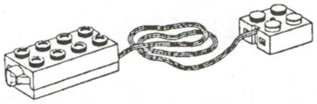
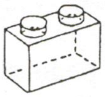
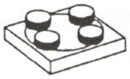
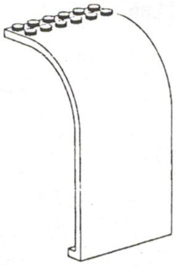
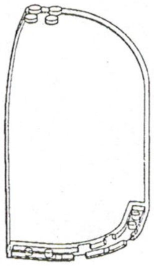
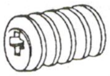
#9701 Technology Set Parts List

Part Name	Quantity
Brick 1x16 red 	1
Brick 2x4 red 	4
Bushing gray 	16
Bushing/axle gray 	6
Chain link black 	54
Connecting lead 5 cm. 	2
Connecting lead 25.6 cm. 	2
Connecting lead 128 cm. 	6
Connector peg black 	24
Corner brick 1x2x2 red 	6
Corner plate 1x2x2 black 	6
Crown gray 	4
Flat tile 1x2 *(four black, six blue, two gray, two red, two white, six yellow)* 	22
Gear 8-tooth gray 	8
Gear 16-tooth gray 	2
Gear 24-tooth gray 	4
Gear rack gray 	14
Gear turntable 	1
Lamp 9v 1x2 white 	4
Lift arm black 	5
Light sensor 9v 	1
Motor 9v gray 	3
Parabolic reflector 	2
Plate 1x2 black 	24
Plate 1x3 black 	4

#9701 Technology Set Parts List

Part Name	Quantity
Plate 1x4 black 	12
Plate 1x6 black 	8
Plate 1x8 black 	9
Plate 2x2 	4
Plate 2x2 with wheel 	1
Plate 2x3 black 	2
Plate 4x4 black 	2
Plate bearing 2x2 red 	2
Plate round 2x2 black 	4
Plate with 1 stud 1x2 black 	4
Plate with holes 2x4 black 	7
Plate with holes 2x8 black 	6
Propeller blade gray 	2
Pulley .250 cm gray 	14
Pulley .875 cm gray 	5
Ridge tile 2x4 black 	1
Roof tile 1x2 red 	6
Roof tile 1x3 inverted red 	4
Roof tile 1x3 red 	2
Rotor black 	2
Sound element 9v 2x2 black 	1
Spiral for connecting leads 	4
Spoked hub *(four gray, four yellow)* 	8
Temperature sensor 9v 	1
Tile 1x4 *(four black, four blue, four yellow)* 	12
Tile 1x8 black 	8

#9701 Technology Set Parts List

Part Name	Quantity
Tile 2x2 round white 	4
Tile 2x2 white 	2
Tile 2x2 white with arrow decal 	1
Tile 2x2 white with ice cream decal 	2
Tire black 	8
Tire O-ring 	4
Touch sensor 9V 	1
Transparent brick 1x2 *(one blue, one green, one red, one yellow)* 	4
Turntable plate 	1
Wall element transparent 5x6x10 	2
Wall element transparent corner 6x6x9 	2
Worm gray 	3

✏ Teacher note:

See the *Reference Guide* For more complete information about these and other Control Lab primitives.

`abs`

Reports the absolute value of its input.

```
Show abs -33
33
```

`alloff (ao)`

Turns off all the output ports except the test port.

```
talkto [motora lampb] on
alloff
```

`and`

Reports true if all its inputs report true; otherwise, it reports false.

```
talkto "motora
if and (light5 > 50) (temp1 > 37)[on]
```

`cleartext`

On a page, clears the text in the current text box. If the Procedures page is currently displayed, `cleartext` clears all the procedures.

```
talkto "text1 print "hello
cleartext
```

`forever`

Runs the input repeatedly as a parallel process. Use the commands `stopall` or `stopme` to stop the forever process.

```
forever [on wait 5 off wait 5]
forever [if temp1 > 75 [talkto
"motora on]]
```

`if`

Runs the instruction list only if the condition reports true.

```
talkto "motora
if angle5 > 50 [onfor 20]
```

`getpage (gp)`

Displays the specified page. The input must be the name of a page in the current project.

```
to slideshow
getpage "page1
wait 30
getpage "page2
wait 30
end
```

`ifelse`

Runs the first instruction list if the condition is true. Runs the second instruction list if the condition is false.

```
talkto "motora
ifelse angle5 > 50 [on] [off]
```

`onfor`

Turns on the current motor, lamp, or sound element for a specified duration. The duration is indicated in tenths of a second.

```
talkto [motora lampb soundc]
onfor 30
```

`output`

Stops the procedure and reports a word or list. `Output` can only be used within a procedure.

```
to rotations
output abs (angle5 / 16)
end

show rotations
```

`print`

Prints a word or list in the current text box or on the Procedures page. *Word-or-list* is followed by a carriage return.

```
talkto "text1
print "hello
```

`resetrotation`

Resets the angle sensor to 0.

```
Show angle6
12

resetrotation 6
show angle6
0
```

`setleft` and `setright`

Sets the direction of motors or sets the sound effect on a sound element.

```
talkto [motora motorb]
setleft
on

talkto "soundc
repeat 10 [setright onfor 20
setleft onfor 20]
```

`setpower`

Sets the power level of the current motor or lamp. The input must be a number between 0 and 8 inclusive.

```
talkto "motora
setpower 5
```

`show`

Prints a word or list in the Command Center. A list is printed with its outer brackets showing.

```
show "hello
hello
show [hello there]
[hello there]
show sentence "|The angle is:| angle6
[The angle is: 180]
```

`stopall`

Stops all running procedures and processes, including button-generated activities, sampling Graph boxes and motors, and returns control directly to the Command Center. The monitors and the Graph boxes remain active.

```
forever [if temp1 > 75 [tto "motora
on]]
stopall
```

`talkto (tto)`

Addresses motors, lamps and sound elements.

```
talkto "motora onfor 10
talkto [motora lampb soundc] on
```

`wait`

Causes a pause in the execution of a program. The time is measured in tenths of one second.

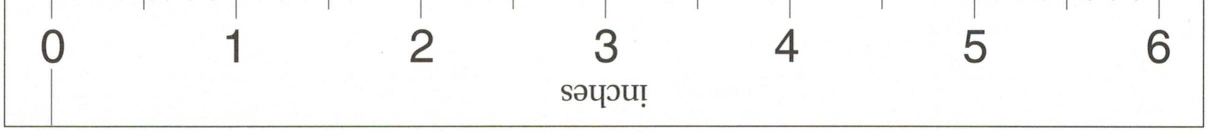
```
repeat 5 [on wait 20 off]
```

`waituntil`

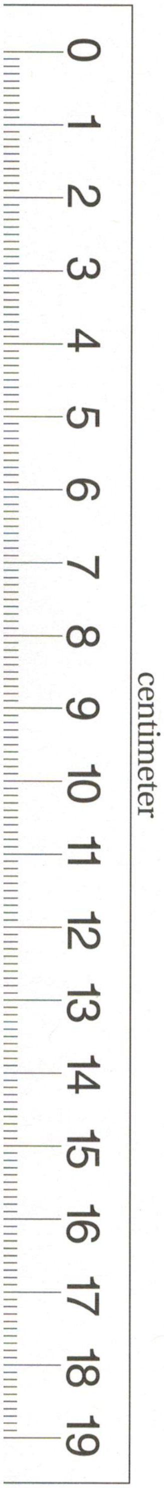
Tells Control Lab to wait until *true-or-false-list-to-run* is true before following another instruction.

```
waituntil [touch2] talkto "motora on
waituntil [not touch2] talkto "motora
off
```

How To	**Mouse Action**
Activate an element.	Click on unselected element; e.g., click on a button to trigger its instruction, click on a text box to activate the cursor in the text box.
Select the element.	Drag around elements or shift-click on elements. The area must include at least one border of the element.
Select or unselect many elements.	Shift-click on elements, shift-click-drag or ⌘-A (Macintosh) or Alt-A (MS-DOS).
Open the dialog box of an element.	Double-click on a selected element or shift-double-click on an element.
Relocate an element.	Shift-drag an element or simply drag the selected element.
Resize an element.	Drag a black handle on a selected element. If the handles are gray, the element can not be resized.
Get Help information about a specific element.	Choose General in the Help menu to select the page element, click Help in the element dialog box or use ⌘-? (Macintosh) or Alt-H (MS-DOS).



English Ruler



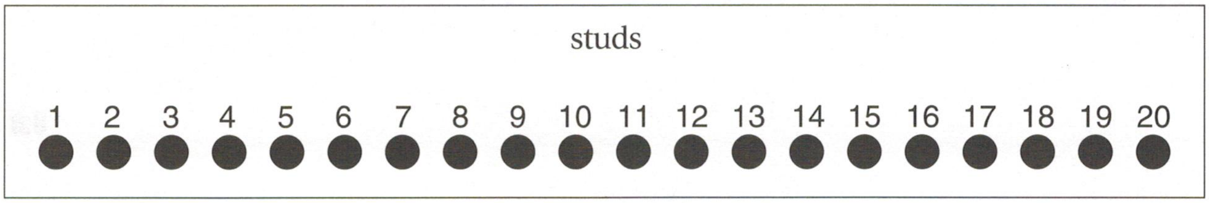
Metric Ruler

Measurement Worksheet

1 inch = 2.54 centimeters
1 inch = 3.20 1 x 1 stud LEGO® bricks
1 centimeter = 0.39 inches
1 centimeter = 1.27 LEGO studs
1 x 1 stud LEGO brick = 5/16 inch
1 x 1 stud LEGO brick = .79 centimeters

LEGO dacta®
The *educational* division of the LEGO Group.

LEGO Ruler



Invention Drawing: Front View

LEGO dacta

The *educational* division of the LEGO Group.

Use this official LEGO® graph paper for designing your models from a FRONT view. Place some LEGO plates (flat pieces) and bricks on this sheet to see how they fit in the lines. Each box on the sheet equals a one-stud plate. Notice that every third line is a little bit darker than the other two. When three plates are stacked together they equal the size of one LEGO brick.

®LEGO and LEGO DACTA are registered trademarks of INTERLEGO AG.

Invention Drawing: Front View

Invention Drawing: Top View

LEGO dacta

The *educational* division of the LEGO Group.

Use this official LEGO® graph paper for designing your models from a TOP view. Place some LEGO plates (flat pieces) and bricks on this sheet with the studs facing upwards. Notice how they fit in the lines. Each box equals a one-stud plate.

®LEGO and LEGO DACTA are registered trademarks of INTERLEGO AG.

Invention Drawing: Top View

Problem Solving Steps

✏ Teacher notes:

This form relates to the "Conceptual Framework" problem solving approach. See the section "The Conceptual Framework Approach" in this book for more information.

Photocopy this form and distribute to students to help them work through the problem solving process in a more organized way.

For a systems approach to problem solving, see the copymasters in this section on Systems and see the section "A Systems Approach" in this book for more information.

Define the Problem

Read the design brief carefully. Identify the important criteria for solving the problem.

Develop Alternative Solutions

1)Sketch or describe at least three designs for the system.
2)Describe at least two ways of controlling the system using LEGO DACTA® Control Lab.

Select a Solution

Review the building and programming features you described above. Choose the best overall solution to build and program.

Develop and Test a Solution

Build and program the solution you selected.

Evaluate the Solution

After you have tested the system, evaluate how well it functions. Here are some ideas to think about.

- Is the structure stable?
- Does the system respond reliably?
- Ask another person to operate the system. Is the control panel easy to understand? What improvements does he or she recommend?
- If appropriate, graph the feedback from the system. Then review the graph showing feedback. How quickly or slowly did the system respond? How could you make the system respond better?

Redesign the Solution

What features of the building solution would you change? Why?

What features of the control solution would you change? Why?

Will these changes make the system more cost effective to produce?

Will these changes make the system safer and easier to operate?

Interpret the Solution

Make the system easy for others to understand and use by adding text boxes and picture shapes to the project pages.

Print the project pages when you are finished.

Prepare a poster or presentation to share with your classmates or family.

Human, Social and Environmental Impacts

Consider the impact of the system you developed on everyday life. Also think about the resources used to develop, operate and maintain the system.

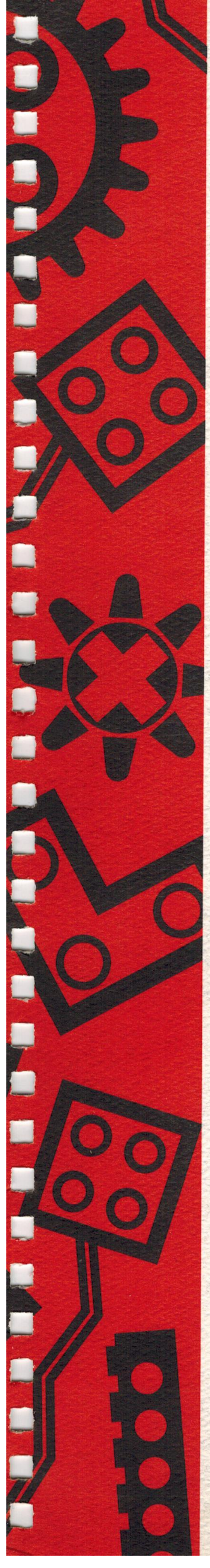
- What impact does the system have on how people live, travel, and work?
- What are potential negative impacts?
- What are potential positive impacts?
- Does the system introduce new environmental hazards that must be monitored? How will you know if the system is safe enough for the environment and people?

Career Related Impacts

What skills are required to design and construct the system?

What skills are required to operate the system?

You are responsible for hiring three people to design and program your system. Write job descriptions for each person. Do they all have the same skills? Will they be working together or separately?



LEGO DACTA®

ControlLAB™

Technology Investigations and Inventions

LEGO dacta®

The *educational* division of the LEGO Group.

Project Evaluation Checklist

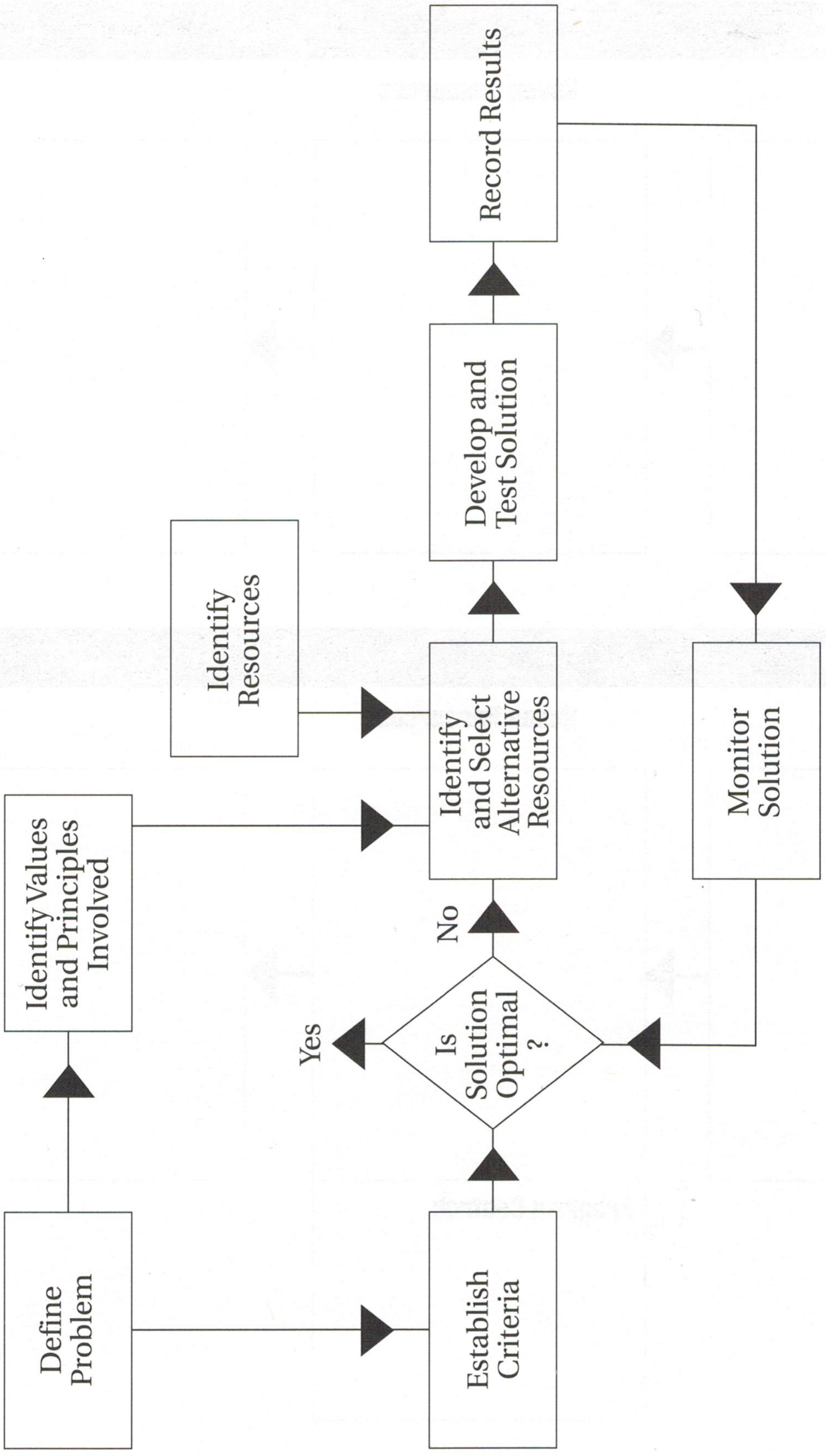
✏ **Teacher note:**
This checklist provides one way of assessing student work. It is especially suitable for use with the "Systems" approach to problem solving.

Student Name:________________________
Class period:________________________
Date:_____________________________

	Possible Points	Points Earned
DOCUMENTATION (60 points)		
Brief and Specifications (10 pts)		
(A) No brief or specifications.	0	
(B) Brief included, no specifications.	5	
(C Brief and specifications included.	10	
Investigation and Research (10 pts)		
(A) No evidence of investigation or research.	0	
(B) Some research with evidence of reading about the problem but lacking depth.	5	
(C) Evidence of good research and that student has acquired new knowledge about the nature of the problem and what others have done to solve similar problems.	10	
Solutions (10 pts)		
(A) Only one solution considered.	0	
(B) Two solutions generated, only one seriously considered.	3	
(C) At least three solutions generated but reasons for rejecting alternatives unclear.	5	
(D) At least three solutions generated and two seriously considered with reasons stated for rejecting alternatives.	8	
(E) At least four solutions generated and three solutions seriously considered. Good reasons for rejected alternatives clearly stated.	10	
Project Testing (10 pts)		
(A) Project not tested.	0	
(B) Project tested but no presentation of results	4	
(C) Project tested with weak presentation of results	7	
D) Project tested with good presentation of results	10	
Evaluation (10 pts)		
(A) No evaluation written.	0	
(B) Poor evaluation with little thought or intelligent comment or containing unjustified self-congratulation	2	

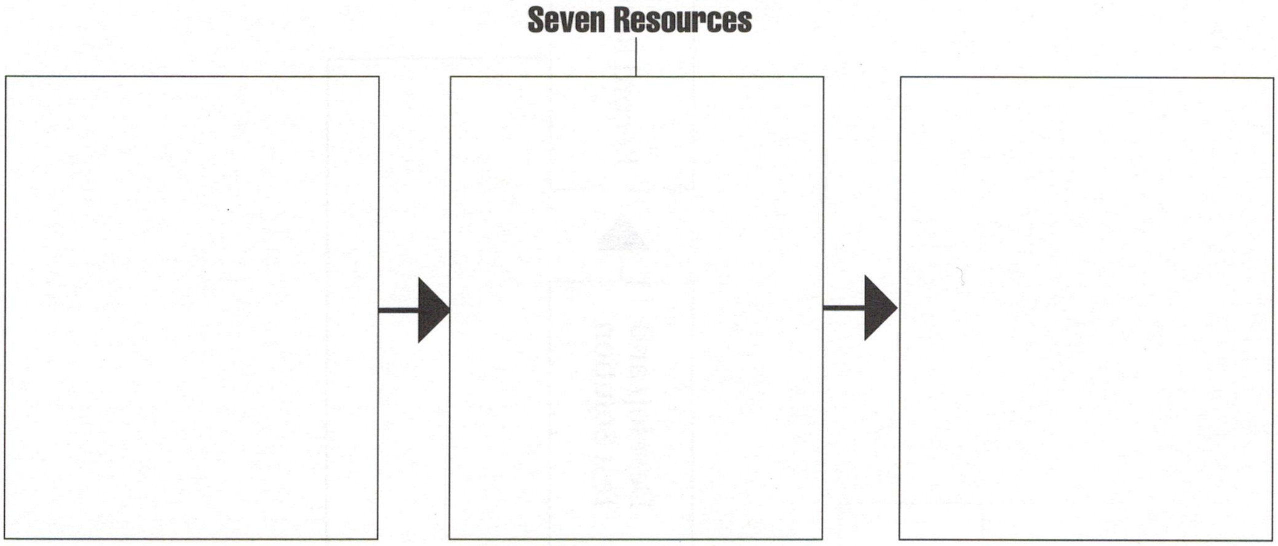
	Possible Points	Points Earned
(C) An evaluation with no self-criticism and little insight.	4	
(D) Good evaluation including self-criticism where appropriate. Possible suggestions about alternatives which, if followed, could lead to improvements to project.	8	
(E) Good evaluation including self-criticism where appropriate. Detailed reasons given for the success of the project OR details given of the modifications or alterations which would ensure a successful conclusion.	10	
Overall Presentation (10 pts)		
(A) No documentation or work	0	
(B) Limited documentation but little illustrative material. Not well organized or presented. Not very neat.	4	
(C) Documented with some illustrative materials. An attempt at organization and reasonably neat in appearance.	7	
(D) Good documentation with good use of illustrative materials. Well organized and neat.	10	
PROJECT (40 points)		
(A) Very poor work, no evidence of craftsmanship or no evidence of any work.	0	
(B) Generally poor work with little craftsmanship shown.	8	
(C) An average amount of work completed but with little care or craftsmanship shown.	16	
(D) Average amount of work with several examples of attention to detail.	24	
(E) Good quality work including realistic choice and use of materials where appropriate. Room for some improvement.	32	
(F) Evidence of thorough work appropriate to time allotted. Construction completed with attention to detail.	40	
TOTAL	**100**	

Systems Problem Solving Model

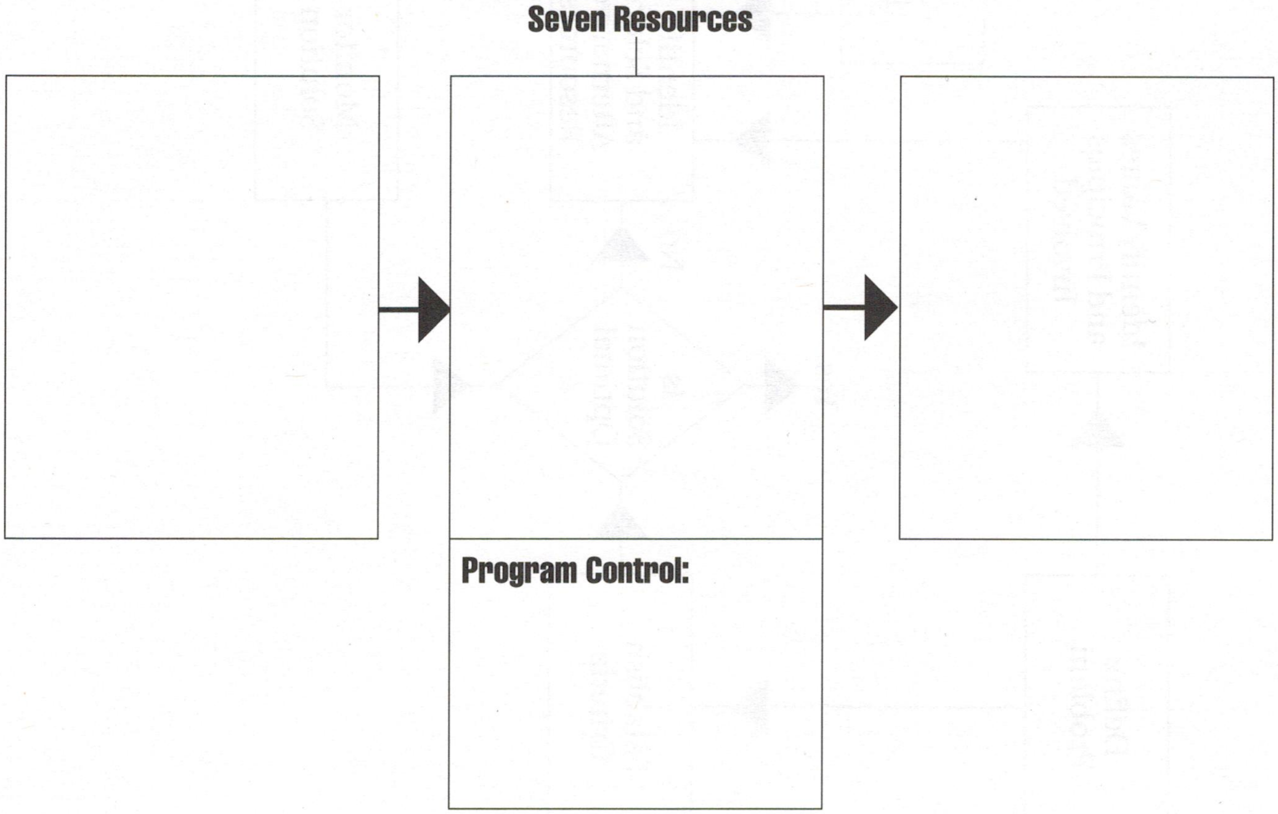


Systems Diagrams: Open Loop and Program Control

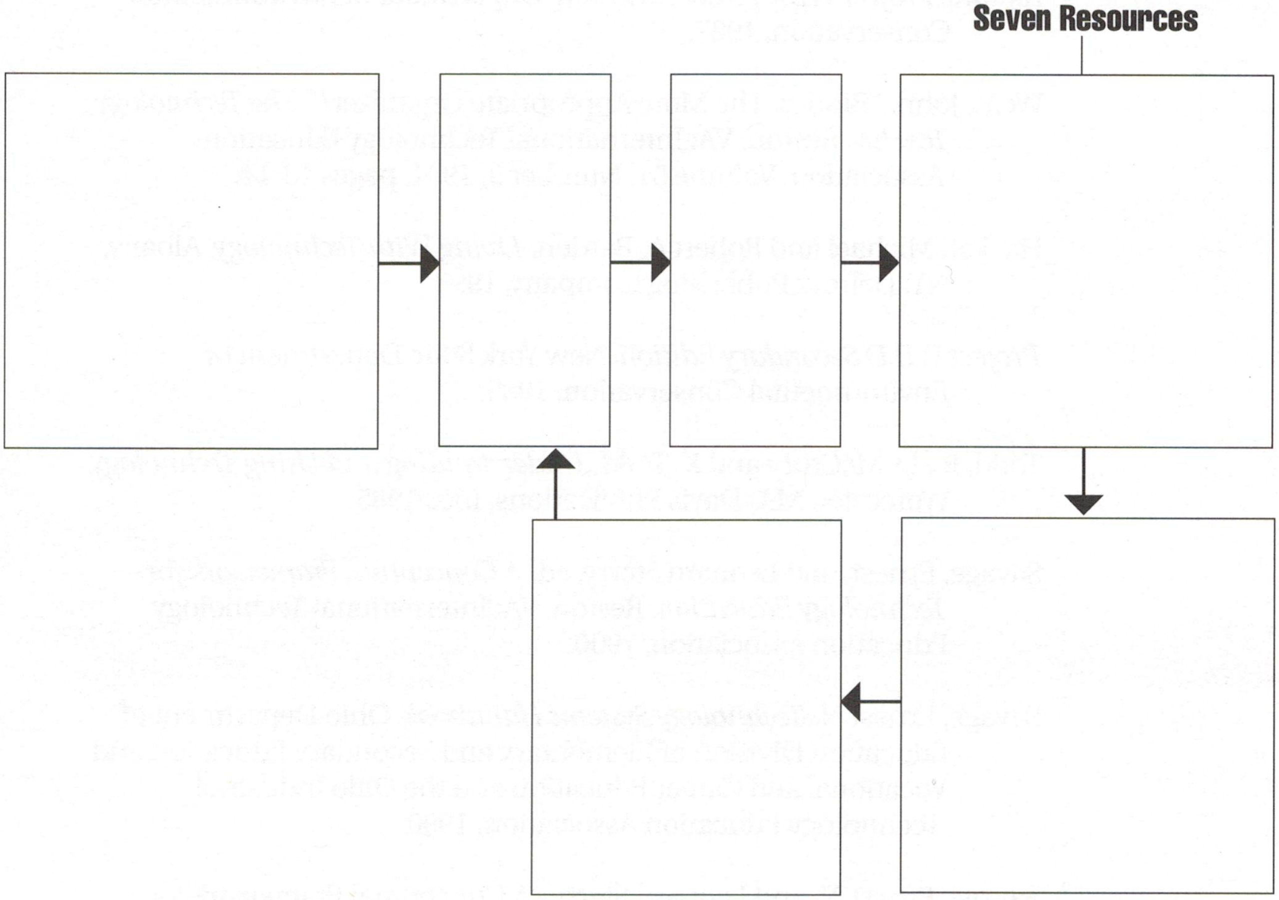
Open Loop



Program Loop



Closed Loop



Bibliography

Technology Education

Aquatic Project WILD. New York State Department of Environmental Conservation, 1987.

Wells, John. "Biotics, The More Appropriate Organizer?" *The Technology Teacher.* Reston, VA: International Technology Education Association, Volume 51, Number 3, 1991, pages 13-14.

Hacker, Michael and Robert A. Barden. *Living With Technology.* Albany, NY: Delmar Publishing Company, 1993.

Project WILD Secondary Edition. New York State Department of Environmental Conservation, 1985.

Todd, R., D. McCrory and K. Todd. *Understanding and Using Technology.* Worcester, MA: Davis Publications, Inc., 1985.

Savage, Ernest. and Leonard Sterry, ed. *A Conceptual Framework for Technology Education.* Reston, VA: International Technology Education Association, 1990.

Savage, Ernest N. *Technology Systems Handbook.* Ohio Department of Education Division of Elementary and Secondary Education and Vocational and Career Education and the Ohio Industrial Technology Education Association, 1990.

Savage, Ernest N. and Leonard Sterry. "A Conceptual Framework for Technology Education." *The Technology Teacher.* Reston, VA: International Technology Education Association, Volume 50, Number 1, 1990, pages 6 - 11.

TechDirections. Ann Arbor, MI: Prakken Publications, Inc.

TIES. Philadelphia, PA: Drexel University.

The Technology Teacher: A Journal of the International Technology Education Association. Reston, VA: ITEA.

General Logo

Burke, Michael, P. and Genise, L. Roland. *Logo and Models of Computation.* Menlo Park, CA: Addison Wesley Publishing Company, 1987.

Friendly, Michael. *Advanced Logo: A Language for Learning.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.

Harper, Dennis. Logo: *Theory and Practice.* Monterey, CA: Brooks / Cole Publishing, 1986.

Harvey, Brian. *Computer Science Logo Style Volumes I-III.* Cambridge, MA: MIT Press, 1985.

Hurley, James. *Logo Physics.* New York, NY: Holt, Rinehart and Winston, 1985.

Logo Update: The Logo Foundation Newsletter. New York: NY.

Tipps, Steve, et. al. *Nudges: IBM Logo Projects.* New York, NY: Holt, Rinehart and Winston, 1984.

Logo Exchange: Journal for Logo-Using Educators. Eugene, OR: International Society for Technology in Education, University of Oregon.

General Education and Other Resources

Caney, Steven. *Steven Caney's Invention Book.* New York, NY: Workman Publishing Company, 1985.

Carnevale, Anthony, P., et al. *Workplace Basics: The Essential Skills Employers Want.* San Francisco, CA: Jossey-Bass Inc. Publishers, 1990.

Educational Leadership: A Journal of the Association for Supervision and Curriculum Development. Alexandria, VA: ASCD.

Fuller, James L. *Robotics: Introduction, Programming, and Projects.* New York, NY: MacMillan Publishing Company, 1991.

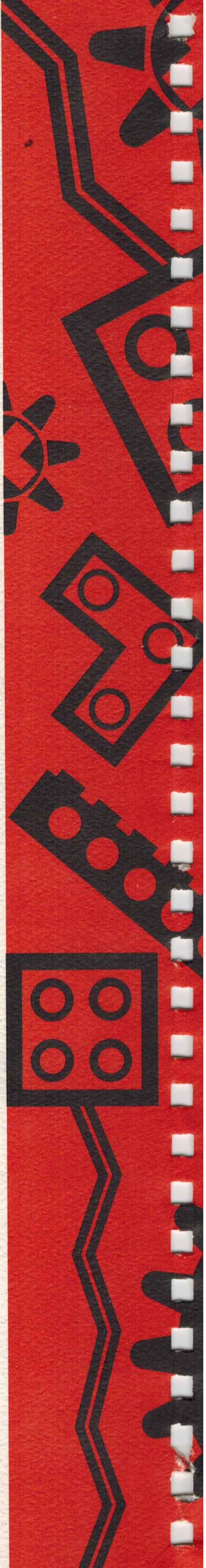
Herman, et al. *A Practical Guide to Alternative Assessment.* Alexandria, VA: Association for Supervision and Curriculum Development, 1992.

Macaulay, David. *The Way Things Work.* Boston, MA: Houghton Mifflin, 1988.

Papert, Seymour. *Mindstorms.* New York, NY: Basic Books, 1980.

Papert, Seymour. *The Children's Machine: Rethinking School in the Age of the Computer.* New York, NY: Basic Books, 1993.

Tippens, Paul E. *Basic Technical Physics.* New York, NY: McGraw-Hill Book Company, 1983.



822206

Printed in the U.S.A.

LEGO DACTA® Control Lab Technology Investigations and Inventions	ISBN 0-914831-90-9
LEGO DACTA® Control Lab Literature Pack	ISBN 0-914831-98-4